

Oracle® Communications Session Border Controller

Accounting Guide



Release S-Cz10.1.0

G49657-03

April 2026

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2026, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About this Guide

My Oracle Support ii

Revision History

1 Accounting on the SBC

| | |
|--------------------------------------|---|
| RADIUS Account Server Prioritization | 1 |
| ACLI Instructions | 2 |
| Create an Account Configuration | 2 |
| Create Accounting Servers | 3 |

2 Call Signaling Accounting Configuration

| | |
|---|----|
| Session Accounting | 1 |
| RADIUS Messages | 1 |
| Session Termination | 2 |
| Interim RADIUS Records for Recursive Attempts | 2 |
| SIP CDR Stop Time | 3 |
| ACLI Instructions and Examples | 3 |
| Media Stop Time VSA in CDRs | 3 |
| Media Stop Time VSAs | 3 |
| Media Stop Time Calculation | 4 |
| Millisecond Granularity for CDRs | 4 |
| Configure acct-session-time for millisecond granularity | 5 |
| SIP Call Tear Down Due to Media Guard Timer Expiration | 5 |
| Set Accounting Message Generation | 5 |
| Enhanced Stop CDR Reporting for Exceeded Ingress Session Constraints | 6 |
| Including P-Visited Network Identifier and History-Info Headers in CDRs | 7 |
| Including the Reason-Header AVP in CDRs | 9 |
| Configure the System to Add Reason Headers | 13 |
| RADIUS CDR Content Control | 13 |
| Configuring Duplicate RADIUS Attribute Generation | 13 |

| | |
|--|----|
| Configuring Specific RADIUS Attributes for CDR Inclusion | 14 |
| Custom RADIUS CDR VSAs for SIP | 15 |
| About User-Defined VSAs for SIP Calls | 15 |
| HMR Adaptations | 15 |
| HMR String Variable | 16 |
| Configure User-Defined VSAs | 16 |
| ACLI Instructions String Variable | 19 |
| Trunk-Group VSA Generation | 19 |
| ACLI Instructions and Examples | 20 |

3 RADIUS Accounting Management

| | |
|----------------------------------|---|
| Alarm Generation and Monitoring | 1 |
| RADIUS Alarms | 1 |
| Status and Statistics Monitoring | 2 |
| ACLI Show RADIUS Display | 2 |

4 Local CDR Files and Push Receivers

| | |
|--|----|
| Local CDR File Format | 1 |
| Local CDR File Format Consistency | 1 |
| Decoding Local CDR File Contents | 2 |
| Generate Local CDR Layout Files | 2 |
| Requirements | 3 |
| Local CDR File Naming Convention | 3 |
| Temporary File Naming for an Open CDR File | 4 |
| Operational Details | 4 |
| HA Considerations for CDR Output Redundancy | 5 |
| Call Detail Record Sequence Number in Filename | 6 |
| Local CDR File Storage Directories | 7 |
| Local CDR File Size and Rotation | 8 |
| Local CDR File Compression | 8 |
| Local CDR File Redundancy | 8 |
| Caveats for H.323 | 8 |
| Configuring Local CDR Files | 8 |
| Local CDR File Delete Warning | 9 |
| Local CDR File Delete Alarm | 10 |
| Local CDR File Delete SNMP Trap | 10 |
| Local CDR Push Receivers | 10 |
| Secure FTP Push Configuration | 10 |
| Multiple Push Receivers | 11 |
| Deprecated ACLI Configuration | 11 |

| | |
|--------------------------------------|----|
| Configuring Local CDR Push Receivers | 12 |
| Monitoring CDR Push Receivers | 13 |

5 CDR Attribute Reference

| | |
|---|----|
| Standard RADIUS Attributes | 1 |
| Standard RADIUS Attributes Dictionary | 2 |
| RADIUS Accounting Termination Causes | 3 |
| Cisco Systems RADIUS Decodes | 3 |
| Oracle RADIUS VSAs | 4 |
| R-Factor and MOS | 5 |
| Media Flow Attributes | 6 |
| IPv6 Support | 25 |
| Oracle VSA Values | 27 |
| Authentication VSAs | 30 |
| RTP Traffic Reporting per Call | 31 |
| VoLTE and SMS VSAs | 32 |
| Distinct VoLTE Processes | 35 |
| Including P-Visited Network Identifier and History-Info Headers in CDRs | 37 |
| P-Asserted-ID Header Format in CDRs | 39 |
| Oracle RADIUS Acme-Extended-Attributes VSAs | 40 |
| MSRP Attributes | 40 |
| STIR/SHAKEN Attributes | 41 |
| Modem Tone Attributes | 45 |
| Mappings and Disconnect Cause Values | 46 |
| SIP H.323 and Q.850 Mappings | 46 |
| SIP Status to H.323 Disconnect Reason Mapping | 46 |
| SIP Status to H.323 RAS Error Mapping | 47 |
| SIP Status to H.323 Release Complete Reason Error Mapping | 47 |
| Q.850 Cause to H.323 Release Complete Reason Mapping | 48 |
| SIP-SIP Calls | 48 |
| SIP-H.323 Calls with Interworking | 48 |
| SIP Events and Errors | 49 |
| H.323 Events and Errors | 49 |
| H.225 RAS Errors | 50 |

6 Diameter Accounting

| | |
|------------------------------|---|
| Diameter Accounting Messages | 1 |
| Resending ACRs | 1 |
| Postponement Feature | 1 |
| Call Flow Examples | 1 |

| | |
|--|----|
| Per Realm Accounting Control | 3 |
| ACLI Instructions | 4 |
| ACR AVP Descriptions | 4 |
| Session-Id AVP (263) | 5 |
| Origin-Host AVP (264) | 5 |
| Origin-Realm AVP (296) | 5 |
| Destination-Realm AVP (283) | 5 |
| Destination-Host AVP (293) | 5 |
| Accounting-Record-Type AVP (480) | 5 |
| Accounting-Record-Number AVP (485) | 5 |
| Acct-Application-Id AVP (259) | 7 |
| User-Name AVP (1) | 7 |
| Event-Timestamp AVP (55) | 7 |
| Event-Type AVP (823) | 7 |
| Role-of-Node AVP (829) | 7 |
| User-Session-Id AVP (830) | 8 |
| Calling-Party-Address AVP (831) | 8 |
| Called-Party-Address AVP (832) | 8 |
| Acme-SipHdr-TO AVP (122) | 8 |
| Time-Stamps AVP (833) | 8 |
| Inter-Operator-Identifier AVP (838) | 9 |
| SDP-Session-Description AVP (842) | 9 |
| Session-Media-Component AVP (845) | 9 |
| Cause AVP (860) | 9 |
| Values for Cause Code AVP (861) | 9 |
| Acme-VBDTone-Detected (162) | 10 |
| STIR/SHAKEN AVPs | 10 |
| Stir-Signed-Request AVP (104) | 11 |
| Stir-Signed-Request-Exception-Id AVP (105) | 11 |
| Stir-Verified-Request AVP (106) | 12 |
| Stir-Verified-Request-Exception-Id AVP (107) | 12 |
| Stir-VS-Verstat (108) | 13 |
| Stir-VS-Reason (109) | 13 |
| Stir-TN-Used-For-AS-VS-Request (110) | 14 |
| Stir-Div-Signed-Request (111) | 14 |
| Stir-Div-Verified-Request (112) | 14 |
| Stir-VS-Invite-State (118) | 15 |
| Stir-VS-attest (119) | 15 |
| Stir-Div-VS-Reason (120) | 15 |
| Stir-Div-VS-Verstat (121) | 16 |
| Stir-AS-Bypass-Token | 16 |
| Acme-P-VSA-NNN (123-153) | 16 |

| | |
|---|----|
| ACR Event Records | 17 |
| ACR Event Message Construction | 17 |
| Event-Type AVP | 17 |
| Expires Value | 18 |
| Event ACRs Generated for Unsuccessful Session Attempts | 18 |
| Event ACRs Generated for Receipt of SIP Messages | 20 |
| Event ACRs for SMS | 21 |
| VoLTE Call and SMS AVPs for Diameter | 23 |
| Distinct VoLTE Processes | 26 |
| Event Local CSV File | 28 |
| Diameter Heartbeat for Rf | 29 |
| Using FQDNs to Access CCFs over Diameter | 29 |
| Handling Multiple CCFs | 32 |
| Accounting Traffic and Server Statistics | 36 |
| Configuring Diameter-based Accounting | 39 |
| Configure the Global Diameter-based Accounting (Rf) Features | 39 |
| Configure Accounting Servers | 42 |
| Create a Dictionary File for Decoding AVPs | 44 |
| Additional Rf Features | 45 |
| Service-Context-ID Format | 45 |
| Acme Excluded Attribute Range | 45 |
| Configure Account | 46 |
| Including the To Header in ACRs and CDRs | 46 |
| Supporting IOI AVPs for Unregistered Endpoints | 47 |
| Managing IOI for Peering Endpoints | 48 |
| Including Number Portability Routing Information in ACRs and CDRs | 52 |
| SNMP Trap Behavior | 53 |
| Alarms | 55 |
| SNMP MIBs and Traps | 55 |
| ApDiamResultCode Textual Convention | 55 |
| apDiameterSvrErrorResultTrap | 56 |
| apDiameterSvrSuccessResultTrap | 56 |
| apDiamACCTResultObjectsGroup Object Group | 57 |
| apDiamACCTResultNotificationsGroup Notification Group | 57 |
| SNMP Varbind Definitions | 57 |
| Diameter Rf Charging Buffering and Storage | 57 |
| About Buffering | 57 |
| About Storage | 58 |
| Monitoring Storage Space | 58 |
| ACLI Instructions and Examples | 58 |
| SNMP | 59 |
| DIAMETER Rf Charging Failure & Recovery Detection | 59 |

| | |
|----------------------------------|----|
| Associated Traps | 59 |
| Custom Diameter ACR AVPs for SIP | 60 |

A RADIUS Dictionary Reference

B Local CDR Table Layouts

| | |
|---|------|
| Local CDR Start Record (RADIUS) | B-1 |
| Local CDR Interim Record (RADIUS) | B-5 |
| Local CDR Stop Record (RADIUS) | B-9 |
| Local CDR Message Record for SMS (RADIUS) | B-15 |
| Local CDR Start Record (Diameter) | B-16 |
| Local CDR Interim Record (Diameter) | B-20 |
| Local CDR Stop Record (Diameter) | B-26 |
| Local CDR Message Record for SMS (Diameter) | B-32 |

C Oracle Rf Interface Support

| | |
|---|------|
| Diameter AVP Notation | C-1 |
| Table Explanation | C-1 |
| Root ACR Message Format | C-1 |
| Service Information AVP | C-2 |
| Subscription ID AVP | C-2 |
| IMS Information AVP | C-2 |
| Reason-Header AVP (3401) | C-3 |
| Number-Portability-Routing-Information (2024) | C-4 |
| Event-Type AVP | C-4 |
| Time Stamps AVP | C-5 |
| Inter-Operator-Identifier AVP | C-5 |
| SDP-Media-Component AVP | C-5 |
| Early-Media-Description AVP | C-5 |
| SDP-Timestamps AVP | C-6 |
| Message-Body AVP | C-6 |
| Acme-Packet-Specific-Extension-Rf AVP | C-6 |
| AVP Definitions | C-11 |
| System Alarming Based on Received Result-Code (268) AVP | C-13 |
| Interim ACR Message Creation Interval per Acct-Interim-Interval AVP | C-14 |

About this Guide

Overview

The Oracle Communications Session Border Controller Accounting Guide describes:

- A description of how to high level accounting configuration and how to configure one or more remote RADIUS servers, including a failover group of RADIUS serves
- A description of how RADIUS CDRs are populated for H.323 and SIP Calls, including accounting-specific features
- A brief section on managing the RADIUS accounting servers, including alarms and statistics
- Local CDR file format and push receivers
- A section on CDR attribute details
- Diameter-based Rf Accounting

The appendicies in this publication include

- Inclusive local CDR layouts for generic calls, and at the 3 points of generation (stop, interim, start)
- Diameter Rf specification for supported AVPs

Documentation Set

The following table describes the documentation set for this release.

| Document Name | Document Description |
|--|---|
| Acme Packet 3900 Hardware Installation Guide | Contains information about the components and installation of the Acme Packet 3900. |
| Acme Packet 4600 Hardware Installation Guide | Contains information about the components and installation of the Acme Packet 4600. |
| Acme Packet 4900 Hardware Installation Guide | Contains information about the components and installation of the Acme Packet 3950 and Acme Packet 4900. |
| Acme Packet 6350 Hardware Installation Guide | Contains information about the components and installation of the Acme Packet 6350. |
| Acme Packet 6400 Hardware Installation Guide | Contains information about the components and installation of the Acme Packet 6400. |
| Release Notes | Contains information about the current documentation set release, including new features and management changes. |
| Known Issues & Caveats | Contains known issues and caveats |
| Configuration Guide | Contains information about the administration and software configuration of the Service Provider Session Border Controller (SBC). |

| Document Name | Document Description |
|---|---|
| ACLI Reference Guide | Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters. |
| Maintenance and Troubleshooting Guide | Contains information about SBC logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives. |
| MIB Guide | Contains information about Management Information Base (MIBs), Oracle Communication's enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects. |
| Accounting Guide | Contains information about the SBC's accounting support, including details about RADIUS and Diameter accounting. |
| HDR Guide | Contains information about the SBC's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information. |
| Admin Security Guide | Contains information about the SBC's support for its Administrative Security license. |
| Security Guide | Contains information about security considerations and best practices from a network and application security perspective for the SBC family of products. |
| Platform Preparation and Installation Guide | Contains information about upgrading system images and any pre-boot system provisioning. |
| Call Traffic Monitoring Guide | Contains information about traffic monitoring and packet traces as collected on the system. This guide also includes WebGUI configuration used for the SIP Monitor and Trace application. |
| HMR Guide | Contains information about configuring and using Header Manipulation Rules to manage service traffic. |
| REST API | Contains information about the supported REST APIs and how to use the REST API interface. |

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support (CAS) can assist you with My Oracle Support registration.

Call the CAS main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select 2 for New Service Request.
2. Select 3 for Hardware, Networking, and Solaris Operating System Support.

3. Select one of the following options:
 - For technical issues such as creating a new Service Request (SR), select 1.
 - For non-technical issues such as registration or assistance with My Oracle Support, select 2.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Emergency Response

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system's ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.

Locate Product Documentation on the Oracle Help Center Site

Oracle Communications customer documentation is available on the web at the Oracle Help Center (OHC) site, <http://docs.oracle.com>. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at <http://www.adobe.com>.

1. Access the Oracle Help Center site at <http://docs.oracle.com>.
2. Click **Industries**.
3. Under the Oracle Communications sub-header, click the **Oracle Communications documentation** link.

The Communications Documentation page appears. Most products covered by these documentation sets appear under the headings "Network Session Delivery and Control Infrastructure" or "Platforms."
4. Click on your Product and then Release Number.

A list of the entire documentation set for the selected product and release appears.
5. To download a file to your location, right-click the **PDF** link, select **Save target as** (or similar command based on your browser), and save to a local folder.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Revision History

This section provides a revision history for this document.

| Date | Description |
|------------|--------------------|
| April 2026 | • Initial release. |

1

Accounting on the SBC

RADIUS is an accounting, authentication, and authorization (AAA) system. In general, RADIUS servers are responsible for receiving user connection requests, authenticating users, and returning all configuration information necessary for the client to deliver service to the user. This document focuses on capturing call accounting data.

You can configure your SBC to send call accounting information to one or more RADIUS servers. This information can help you to see usage and QoS metrics, monitor traffic, and even troubleshoot your system. For more information about QoS, refer to the Admission Control and QoS chapter of the Configuration Guide.

Accounting data may also be written locally in a friendly CSV format in standard text files. You can automate systems to retrieve these files from the SBC's directories, or you can configure the SBC to SFTP the files at regular intervals to remote servers within your network.

Finally, accounting information may be relayed to servers, often in an VoLTE network, using the Rf charging interface which runs on the Diameter protocol. The SBC can output either RADIUS or Diameter based accounting information, but not both simultaneously.

Entitlement

In order to use RADIUS with your SBC, you must have the Accounting entitlement installed and activated on your system. For more information, see the Getting Started chapter of the *Configuration Guide*.

Accounting Features

For H.323, SIP, and calls being interworked between H.323 and SIP (IWF), you can obtain records that contain information to help you with accounting and that provide a quantitative and qualitative measurement of the call. For H.323 and SIP calls, the SBC generates one set of records; for calls requiring IWF, the SBC generates two sets of records.

You can use the RADIUS records generated by your SBC to assist you with:

- Usage accounting—See the calling and called parties for a call, the protocol used, the realm the call traversed (as well as local and remote IP address and port information), and the codec used
- Traffic monitoring—You can see information about the setup, connect, and disconnect times, as well as the SIP or H.323 disconnect cause
- SLA monitoring—The SBC supports RADIUS attributes that provide information about jitter, latency, and loss for H.323, SIP, and calls that require interworking between H.323 and SIP
- Troubleshooting—Obtain information about calls that can help you to identify and address issues with quality and how calls are setup and torn down.

RADIUS Account Server Prioritization

Especially useful for customers with multiple SBCs, the RADIUS account server prioritization feature allows you to assign a priority to each of the account servers you configure. Setting the priority for RADIUS accounting servers allows you to load balance traffic across the servers.

Without this feature, the SBC sorts RADIUS accounting servers by their IP addresses and ports. For example, if you have a pre-existing accounting server with the IP address and port combination of 10.1.31.2:1813 and then configure a new server at 10.0.3.12:2145, the new server will take priority over the pre-existing one. Of course, you always have the option of allowing the system to set the priority or your accounting servers in this way.

The prioritization feature works with all of the strategy types you set in the accounting configuration. However, it is most applicable to the **hunt** or **failover** strategies. You can assign a number to each server to mark its priority, or you can leave the priority parameter set to 0 (default) so the SBC prioritizes them by IP address and port.

ACLI Instructions

This section tells you how to access and set parameters for accounting support. To use the SBC with external RADIUS (accounting) servers to generate CDRs and provide billing services, you need to configure account configuration and one or more account servers.

Create an Account Configuration

You set the account configuration parameters to define high level accounting information. After setting initial parameters in the account-config, you must create one or more accounting-servers that define the systems where RADIUS accounting CDRs are sent.

To configure the account configuration:

1. Access the **account-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

2. **hostname**—Defaults to and must remain localhost.
3. **port**—Retain the default value of 1813 or enter the number of the UDP port associated with the SBC from which RADIUS messages are sent.
 - minimum: 1025
 - maximum: 65535
4. **strategy**—Indicate the strategy you want used to select the accounting servers to which the SBC will send its accounting messages. The following table lists the available strategies:
 - **hunt**—Selects accounting servers in the order in which they are listed. If the first accounting server is online, working, and has not exceeded any of the defined constraints, all traffic is sent to it. Otherwise the second accounting server is selected. If the first and second accounting servers are offline or exceed any defined constraints, the third accounting server is selected. And so on through the entire list of configured servers.
 - **failover**—Uses the first server in the list of predefined accounting servers until a failure is received from that server. Once a failure is received, it moves to the second accounting server in the list until a failure is received. And so on through the entire list of configured servers.
 - **round robin**—Selects each accounting server in order, distributing the selection of each accounting server evenly over time.

- fastest round trip time—Selects the accounting server that has the fastest round trip time (RTT) observed during transactions with the servers (sending a record and receiving an ACK).
 - fewest pending—Selects the accounting server that has the fewest number of unacknowledged accounting messages (that are in transit to the SBC).
5. **protocol**—Retain the default value to use RADIUS accounting, or change this to **diameter**, for Diameter Rf charging.
 6. **state**—Retain the default value **enabled** if you want the account configuration active on the system. Enter **disabled** if you do not want the account configuration active on the system.
 7. **max-msg-delay**—Retain the default value of **60** seconds or indicate the length of time in seconds that you want the SBC to continue trying to send each accounting message. During this delay, the SBC can hold a generic queue of 4096 messages.
 - Minimum: zero (0)
 - Maximum: 4294967295
 8. **max-wait-failover**—Retain the default value of **100** messages or indicate the maximum number of accounting messages the SBC can store its message waiting queue for a specific accounting server, before it is considered a failover situation.

Once this value is exceeded, the SBC attempts to send its accounting messages, including its pending messages, to the next accounting server in its configured list.

 - Minimum: one (1) message
 - Maximum: 4096 messages
 9. **trans-at-close**—Retain the default value of **disabled** if you do not want to defer the transmission of message information to the close of a session. Enter **enabled** if you want to defer message transmission.
 - **disabled**—The SBC transmits accounting information at the start of a session (Start), during the session (Interim), and at the close of a session (Stop). The transmitted accounting information for a single session might span a period of hours and be spread out among different storage files.
 - **enabled**—Limits the number of files on the SBC used to store the accounting message information for one session. It is easiest to store the accounting information from a single session in a single storage file.
 10. Type **done** to save your configuration.

Continue to the next section to create accounting servers.

Create Accounting Servers

You must establish the list of servers which the SBC sends accounting messages. Create the account server list to store accounting server information for the account configuration. Each account server can hold 100 accounting messages. RADIUS will not work if you do not enter one or more servers in a list.

1. Access the **account-server** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
```

```
ORACLE(account-config)# account-server
ORACLE(account-server)#
```

2. **hostname**—Add the host associated with the account server as an IP address.
3. **port**—Retain the default 1813 or enter the number of the UDP port associated with the account server to which RADIUS messages are sent.
 - minimum: 1025
 - maximum: 65535
4. **state**—Retain the default enabled to enable the account servers on the system or enter disabled to disable them.
5. **min-round-trip**—Retain the default 250 milliseconds or indicate the minimum round trip time of an accounting message.
 - minimum: 10 milliseconds
 - maximum: 5000 milliseconds

A round trip consists of the following:

- The SBC sends an accounting message to the account server.
- The account server processes this message and responds back to the SBC.

If the fastest RTT is the strategy for the account configuration, the value you enter here can be used to determine an order of preference (if all the configured account servers are responding in less than their minimum RTT).

6. **max-inactivity**—Retain the default 60 seconds or indicate the length of time in seconds that you want the SBC with pending accounting messages to wait when it has not received a valid response from the target account server.
 - minimum: 1 second
 - maximum: 300 seconds

Once this timer value is exceeded, the SBC marks the unresponsive account server as disabled in its failover scheme. When a server connection is marked as inactive, the SBC attempts to restart the connection and transfers pending messages to another queue for transmission. RADIUS messages might be moved between different account servers as servers become inactive or disabled.

7. **restart-delay**—Retain the default 30 seconds or indicate the length of time in seconds you want the SBC to wait before resending messages to a disabled account server.
 - minimum: 1 second
 - maximum: 300 seconds
8. **bundle-vsa**—Retain the default enabled if you want the account server to bundle the VSAs within RADIUS accounting messages. Enter disabled if you do not want the VSAs to be bundled. (Bundling means including multiple VSAs within the vendor value portion of the message.)

In a bundled accounting message, the RADIUS message type is vendor-specific, the length is determined for each individual message, and the vendor portion begins with a 4-byte identifier, and includes multiple vendor type, vendor length, and vendor value attributes.

9. **secret**—Enter the secret passed from the account server to the client in text format. Transactions between the client and the RADIUS server are authenticated by the shared secret; which is determined by the source IPv4 address of the received packet.

- 10. NAS-ID**—Optional. Enter the NAS ID in text format (FQDN allowed). The account server uses this value to identify the SBC for the transmittal of accounting messages.

The remote server to which the account configuration sends messages uses at least one of two potential pieces of information for purposes of identification. The SBC accounting messages always includes in the first of these:

- Network Access Server (NAS) IP address (the IP address of the SBC's SIP proxy)
- NAS ID (the second piece of information) provided by this value. If you enter a value here, the NAS ID is sent to the remote server.

If you have more than one SBC pointing to the same account server, the NAS ID can be used to identify which SBC generated the record.

- 11. priority**—Enter the number corresponding to the priority you want this account server to have in relation to the other account servers to which you send traffic. The default for this parameter is 0, meaning the prioritization feature is turned off—and that the SBC will therefore prioritize accounting servers by IP address and port.

2

Call Signaling Accounting Configuration

This section explains SIP and H.323 accounting using the RADIUS Accounting System (RAS).

You can configure the SBC to modify and populate CDRs in specific ways. There are options for how a call proceeds with respect to start/stop/interim records, and the information contained within. This chapter lists features relating to your ability to customize the behavior of call data record population.

Session Accounting

The RAS client can record SIP, H.323, and IWF session activity based on configuration and a CDR. The CDR determines which messages are generated and determines the RADIUS attributes included in the messages. The RAS client must be capable of sending CDRs to any number of RADIUS accounting servers, using the defined hunt, failover, round robin, fewest pending, or fastest server strategies.

The establishment, failed establishment, change, or removal of a session can trigger RADIUS Accounting Request messages. The RAS might also send notification of its status (enabled/disabled). RADIUS Accounting Request messages include the following:

- Start—Session has started.
- Interim-Update—Session parameters have changed.
- Stop—Session has ended.
- Accounting-On—Creation of a new RADIUS client.
- Accounting-Off—RADIUS client has shut down.

Each session might generate Start, Interim-Update, and Stop messages based on the local configuration when the session is initiated. Each Start message tells the RADIUS server that a session has started. Each Interim-Update message changes the session parameters, and may report the session characteristics for the session to that point. Each Stop message informs the RADIUS server that a session has ended and reports session characteristics.

The RAS has the ability to transmit all RADIUS messages related to a session at the end of the session, regardless of which messages are generated and when they are generated. Some customers might choose this option to reduce the likelihood of the RADIUS messages being logged to different servers, or in different log files on the same server.

The RAS always generates a RADIUS Stop message when the session ends, regardless of the session termination cause. The termination cause and the session characteristics are reported.

RADIUS Messages

The following table identifies the relationship between the signaling elements and the RADIUS attributes included in Accounting Request messages to the RADIUS server.

| RADIUS Attribute | Data Element | Message |
|----------------------|---|--------------------------------------|
| NAS IP-Address | IP address of the SIP proxy or the H.323 stack's call signal address. | Start, Interim-Update, Stop, On, Off |
| NAS Port | SIP proxy port or the H.323 stack's call signaling RAS port. | Start, Interim-Update, Stop, On, Off |
| NAS Identifier | Value, if any, set in the optional NAS-ID field for the accounting server that you configure as part of the accounting configuration. This identifier sets the value that the remote server (the accounting server) uses to identify the SBC so that RADIUS messages can be transmitted. The remote server to which the accounting configuration will send messages uses at least one of two pieces of information for identification: NAS IP address: always included in the accounting message NAS identifier: configured in the NAS-ID parameter of the accounting server; if configured, the NAS identifier is sent to the remote server This attribute only appears if a value is configured in the NAS-ID field. | Start, Interim-Update, Stop, On, Off |
| Acct-Session-ID | Either the Call-ID field value of the SIP INVITE message, the callIdentifier of the H.323 message, or RADIUS client information. | Start, Interim-Update, Stop, On, Off |
| Called Station ID | To field value of the SIP INVITE message (a type of message used to initiate a session) or the calledPartyNumber of the H.323 message. | Start, Interim-Update, Stop |
| Calling Station ID | From field value of the SIP INVITE message or the callingPartyNumber of the H.323 message. | Start, Interim-Update, Stop |
| Acct-Terminate-Cause | Reason for session ending (refer to Session Termination session). | Stop, Off |
| Acct-Session-Time | Length of session (time in seconds, or milliseconds if so configured). | Interim-Update, Stop, Off |

Session Termination

Sessions are terminated for reasons that include normal termination, signaling failure, timeout, or network problems. The following table maps RADIUS accounting termination cause codes to network events.

| RADIUS Termination Cause | Event | Message |
|--------------------------|---|---------|
| User request | SIP BYE message or H.323 | Stop |
| User error | SIP signaling failed to establish session (accompanied by disconnect cause) | Stop |
| NAS request | RADIUS server disabled | Off |

Interim RADIUS Records for Recursive Attempts

When the SBC routes calls, it performs local policy look-ups that can return several next hops, ordered by preference. This can also happen as a results of an LRT lookup, an ENUM query

response, or SIP redirect. To set up sessions, the SBC uses—in ordered preference—and recurses through the list if it encounters failures.

You can configure SIP accounting to send RADIUS Interim records when the SBC encounters these failures. The interim message contains: the destination IP address, the disconnect reason, a timestamp for the failure, and the number that was called. This feature is enabled by setting the `generate-interim` parameter to **unsuccessful-attempt**. Please refer to Appendix B to view the format of an unsuccessful-attempt interim record.

SIP CDR Stop Time

You can set up your global SIP configuration so the disconnect time reflected in a RADIUS CDR is the time when the SBC receives a BYE. Enabling this parameter also means the disconnect time is defined when the SBC sends a BYE to the UAS and UAC. Otherwise, the CDR's value is based on when the 200 OK confirms the BYE.

The applicable RADIUS CDR in this case is the standard RADIUS attribute `Acct-Session-Time`, number 46.

ACLI Instructions and Examples

To enable definition of the disconnect time based on the BYE:

1. Access the **sip-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-config
ORACLE(sip-config)#
```

2. Select the **sip-config** object to edit.

```
ORACLE(sip-config)# select
```

```
ORACLE(sip-config)#
```

3. **set-disconnect-time-on-bye**—Set this parameter to **enabled** if you want to use the BYE message as the defining factor for the disconnect time. This parameter is disabled by **default**.
4. Type **done** to save your configuration.

Media Stop Time VSA in CDRs

An accurate portrayal of a call's media stop time is important for billing accuracy. Calls are often terminated well after the media has stopped flowing for such reasons as network or equipment peculiarities.

Media Stop Time VSAs

To record the actual media stop time, the Oracle Communications Session Border Controller writes the following four VSAs in CDR Stop Records:

```
Acme-Flow-Calling-Media-Stop-Time_FS1
Acme-Flow-Called-Media-Stop-Time_FS1
```

Acme-Flow-Calling-Media-Stop-Time_FS2
Acme-Flow-Called-Media-Stop-Time_FS2

These VSAs correspond to:

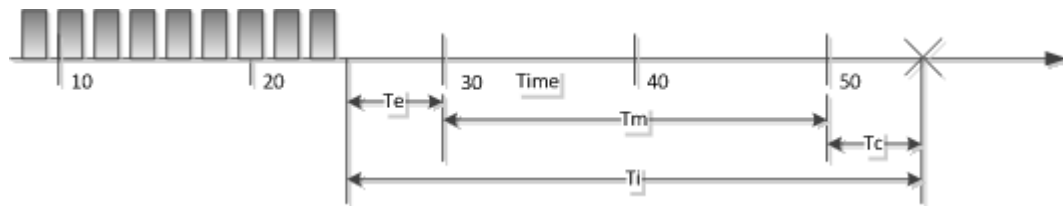
- calling side's media stop time - stream 1
- called side's media stop time - stream 1
- calling side's media stop time - stream 2
- called side's media stop time - stream 2

Media Stop Time Calculation

The granularity of the time at which the Oracle Communications Session Border Controller's checks for media stream idleness, the actual media stop time, as inserted into a CDR is accurate to between 0 and +10 seconds.

In the following diagram, media idleness monitoring is checked in 10 second time frames. Labeled time measurements are as follows:

- T_c —Time between most recent idleness sample end and end-of-call time
- T_m —2 complete idleness windows
- T_e —Time into the idleness window in which the call's media stopped. This is also the error amount of the recorded media stop time
- T_i —The actual time between the end of media and the end of call



T_m and T_c are known. The Oracle Communications Session Border Controller also knows that the media ended between 20 and 30 seconds, but the actual time, $10 - T_e$ into the frame is unknown. Thus, the time recorded in the CDR is quantized up to the end of the media stop frame at 30 seconds. This time, as written to the CDR, must be interpreted with possible error of $0 \leq T_e < 10$ seconds.

Media Stop Time during HA Switchover

When a switchover occurs between media stop time and end of call, the media stop time written to the CDR is the failover time.

Millisecond Granularity for CDRs

Some accounting features require greater precision. The attribute **acct-session-time** can be configured to be in milliseconds.

The RADIUS attribute **acct-session-time** uses seconds as its default. You can set this to a millisecond granularity in the **account-config** configuration element using the option **millisecond-duration**. This option setting is required for the RADIUS CDR display, Diameter RF accounting and locally-generated CDR comma separated value (CSV) files behaviors.

Note

Changing to millisecond granularity violates RFC 2866.

Configure acct-session-time for millisecond granularity

Set the option for millisecond granularity for the **acct-session-time** attribute.

1. Access the **account-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

2. Type **select** to begin configuring this object.
3. **options**—Set the **options** parameter by typing **+options**, a Space, the option name **millisecond-duration** and then press Enter.
4. Type **done** to save your configuration.

SIP Call Tear Down Due to Media Guard Timer Expiration

When a SIP call is torn down by the SBC due to media timers expiring, the following standard and VSA attributes and their corresponding values will appear in the CDR stop message together:

Table 2-1 Media Guard Timer Expiration in CDR

| CDR Attribute | Value | Explanation |
|---------------------------|--------------|---|
| Acct-Terminate-Cause | Idle-Timeout | This standard RADIUS AVP indicates the call was ended due to a timer expiring. |
| h323-disconnect-cause | "6" | This VSA AVP indicates the call was ended due to a timeout. |
| Acme-Disconnect-Initiator | 3 | This VSA AVP indicates the call disconnect was initiated internally from the SBC, and not from an endpoint or due to an unknown reason. |

Set Accounting Message Generation

You set the account configuration parameters to define high level accounting information. After setting initial parameters in the account-config, you must create one or more accounting-servers that define the systems where RADIUS accounting CDRs are sent.

To configure the account configuration:

1. Access the **account-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
```

```
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

2. **generate-start**—Retain the default value **ok** if you want the RADIUS Start message to be generated once the SBC receives an OK message in response to an INVITE.

Other options include:

- Invite—RADIUS Start message should be generated once the SBC receives a SIP session INVITE.
- None—RADIUS Start message is not generated.
- ""—When two quotation marks are entered next to each other (empty), behavior is identical to none value.

3. **generate-interim**—Retain the default value **reinvite response** to cause the SBC to transmit a RADIUS Interim message. (A RADIUS Interim message indicates to the accounting server that the SIP session parameters have changed.)

You can select none, one, or more than one of the following values:

- ok—RADIUS Start message is generated when the SBC receives an OK message in response to an INVITE.
 - reinvite—RADIUS Interim message is generated when the SBC receives a SIP session reINVITE message.
 - reinvite response (default)—RADIUS Interim message is generated when the SBC receives a SIP session reINVITE and responds to it (for example, session connection or failure).
 - reinvite cancel—RADIUS Interim message is generated when the SBC receives a SIP session reINVITE, and the Reinvite is cancelled before the SBC responds to it.
 - unsuccessful-attempt—RADIUS Interim message is generated when a SIP session set-up attempt from a preference-ordered list of next-hop destinations is unsuccessful. This can happen when a local policy lookup, LRT lookup, ENUM query response, or SIP redirect returns a preference-ordered list of next-hop destinations. The interim message contains: the destination IP address, the disconnect reason, a timestamp for the failure, and the number that was called.
4. **intermediate-period**—Set the time in seconds between generating periodic interim records. This parameter defaults to zero, which is not a valid value.

Interim record generation for each call type is as follows:

- H.323—The periodic timer (set to the value you specify in the accounting configuration) is dynamically created when the SBC receives a Connect message and an H.323 call answer method is invoked. The SBC deletes the timer when the H.323 session is terminated.
 - SIP—The periodic timer (set to the value you specify in the accounting configuration) is dynamically created when the SBC receives an initial INVITE message. The SBC deletes the timer when the session is terminated.
5. Type **done** to save your configuration.

Enhanced Stop CDR Reporting for Exceeded Ingress Session Constraints

The SIP **enhanced-cdr** option enables consistent generation of RADIUS Stop records on both ingress and egress paths. With **enhanced-cdr** enabled, RADIUS Stop records are generated in response to any ingress path rejection of a dialog creating SIP INVITE request. The contents of RADIUS Stop records are also written to the local CDR files (if enabled).

Use the following command syntax to enable consistent generation of RADIUS Stop records.

```
ORACLE(sip-config)# options +enhanced-cdr
ORACLE(sip-config)#
```

In legacy releases there was an inconsistency in the generation of RADIUS Stop records when calls are rejected for exceeding configured session ingress or egress constraints. On the egress path, legacy releases rejected such calls with a 503 (Service Unavailable) response and the generation of a RADIUS STOP record. On the ingress path, however, while calls were rejected with a 503 response, RADIUS Stop records were not generated.

Including P-Visited Network Identifier and History-Info Headers in CDRs

You can configure the SBC to add fully compliant P-Visited Network Identifier (PVNI) and History-Info (HI) headers in CDRs. You configure this by adding the **pcscf-cdr-compliance** option to the **account-config**, specifying whether you want to include PVNI (**PVNI-pref**), HI (**HI-pref**), or both. The behavior is dependent on the type of call, including Mobile Terminating (MT) and Mobile Originating (MO), information provided by SIP, and whether you are also using an S8HR profile.

The PVNI and HI fields in CDRs may or may not contain data. When configured, the SBC performs processes to determine whether or not to add:

- P-Visited-Network-ID to the applicable CDR field
- History-Info to the applicable CDR field

You configure the **pcscf-cdr-compliance** in the applicable **account-config** to use these processes within your environment.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)# select
ORACLE(account-config)# options +pcscf-cdr-compliance="PVNI-pref
```

If you save and activate this configuration, the SBC enables PVNI CDR population for MT calls. To configure for both PVNI and HI headers, configure the option with both values separated by a comma and enclosed in quotes.

```
ORACLE(account-config)# options +pcscf-cdr-compliance="PVNI-pref,HI-pref"
```

Support for P-Visited-Network-ID Field

For MT calls, the access SBC, deployed as an A-SBC, inserts the PVNI header in CDRs based on the called party registration cache entry (MCC/MNC). If the Called party registration cache does not have a PVNI value, the A-SBC inserts the **network-id** value from the access side (egress realm) **sip-interface** configuration as the PVNI into CDRs.

For both MO and MT and when you configure it to add PVNI to CDRs, the A-SBC checks for an **s8hr-profile** in the same interface:

- If there is an S8HR profile on the access **sip-interface**:
 - If the SBC receives an MCC/MNC from the Rx server, it creates the PVNI header using the called party registration cache entry (MCC/MNC) and adds it to the CDR.

- If the SBC does not receive an MCC/MNC, It checks whether there is a **network-id** value on the access side **sip-interface**:
 1. If so, the SBC creates the PVNI using that **network-id** value.
 2. If not, the SBC uses the **local-mccmnc** value as the PVNI, and adds it to CDR.

Note

If you have not configured the **local-mccmnc** value in your S8HR profile, the SBC uses the default, which is 999999.

- If there is not an S8HR profile on the access **sip-interface**, the SBC checks whether there is a **network-id** value on the access side **sip-interface**. If so, the SBC uses the **network-id** value as the PVNI, and adds it to CDR.
- If both the S8HR and the egress (access) **network-id** are not configured, the SBC checks whether the initial INVITE/MESSAGE comes from a trusted endpoint and contains a PVNI:
 - If so, the SBC relays the PVNI and add to CDR.
 - If not, the SBC leaves the PVNI field empty.

When you have set the **pcscf-cdr-compliance** option to include PVNI, and the SBC is acting as an I-SBC handling MO/MT calls, the SBC uses the following sequence for populating the CDR field:

1. If configured, uses the **network-id** on the ingress **sip-interface** as PVNI.
2. If populated and from a trusted endpoint, uses the PVNI from the initial INVITE.
3. Leaves the PVNI field empty.

Note

This behavior applies to the INVITE or any Re-INVITE.

Support for History-Info Field

For MO calls, if you have configured the HI option in the **account-config**, SBC uses the History-Info(s) received in the initial INVITE replies, including those with 181, 180 or 200 status-codes. The SBC populates the CDR with the last provisional (>100) or final (200) response containing History-Info(s). If History-Info is not available in provisional or final replies, the SBC leaves the History-Info in the CDR empty.

For MT calls, SBC extracts History-Info header(s) from the initial INVITE and adds them to the CDR. If History-Info is not available in the initial INVITE, the SBC leaves the HI field empty.

If there are multiple History-Info headers in the initial INVITE, the SBC concatenates all the history-info headers values, and without exceeding the default or configured CDR field size, adds them to the CDR.

For example, assume INVITE has three History-Info headers in the following order:

1. HI-1 - 100 characters
2. HI-2 - 100 characters
3. HI-3 - 100 characters

By default, the maximum CDR field size is 246. In this case, the SBC includes the first two History-Info headers in their entirety, and truncates HI-3.

Consider the presence of the following HI headers:

- History-Info: <sip:bob@example.com>;index=1
- History-Info: <sip:office@example.com>;index=1.2;mp=1
- History-Info: <sip:office@192.0.2.5>;index=1.2.1;rc=1.2

The SBC populates the History-Info CDR as follows

```
<sip:bob@example.com>;index=1, <sip:office@example.com>;index=1.2;mp=1,
<sip:office@192.0.2.5>;index= 1.2.1;rc=1.2
```

The History-Info2 CDR Field

In addition to the History-Info CDR field, the SBC supports the History-Info2 CDR field to capture call history information. The system populates this field when the number of characters for population exceeds the maximum number of characters supported by the History-Info field. If multiple history-info headers generate text that exceeds 246 characters, the SBC parses the headers and adds any spillover into History-Info2. This extends the maximum number of these characters to 492.

For example, if the system receives the following 6 HI headers:

- History-Info: [sip:+34606550955@10.197.141.69;user=phone?Privacy=none];index=1
- History-Info: [sip:unknown@unknown.invalid;cause=404];index=1.1
- History-Info: [sip:unknown@unknown.invalid;cause=404];index=1.1.1
- History-Info: [sip:unknown@unknown.invalid;cause=404];index=1.1.1.1
- History-Info: [sip:+34606550955@10.197.141.69;user=phone;cause=404?Privacy=none];index=1.1.1.1.1
- History-Info: [sip:+34818281924925024@10.197.141.69;user=phone;cause=404;index=1.1.1.1.1.1]

The system includes the first four HI headers in the History Info attribute, and the last two in the History Info2 attribute.

The maximum number of characters for retaining History-Info data is 492 characters. Should your deployment exceed that number of characters, you can consider setting the **cdr-attrib-size-limit** option in the **sip-config** to enable the system to drop the earliest History Info headers in the list and avoid truncating the last entries.

Including the Reason-Header AVP in CDRs

You can configure the SBC to include reason header AVPs in diameter STOP and EVENT CDRs within the context of a configured **account-config**. When configured, the SBC does this for both cause information it receives from another device and for cause information the system generates itself. To do this, you enable the **add-reason-header** parameter in the **sip-config**. When enabled, the SBC includes the Reason-Header AVP in Diameter Accounting Request (ACRs) for Accounting-Record-Type [STOP/EVENT] CDRs per spec 3GPP TS 32.299 V13.5.0 for both in-dialog and out-of-dialog messages. This is in addition to the parameter's other functions, including adding the SIP reason header to BYE, CANCEL, and SIP Error Responses (4xx, 5xx, 6xx). You can customize the text used in reason headers using the **local-error** parameters in the **local-response-map** that you apply to your target traffic.

The system checks incoming BYE, CANCEL, and SIP Error Responses (4xx, 5xx, 6xx) for reason headers to propagate using either default or customized messaging you have specified in your local response map. These headers would be presented as Q.850 cause and text information. If the incoming message does not include reason headers, the SBC does not insert a reason header AVP in the ACR or a reason header in the ensuing signaling. If it receives multiple reason-headers, the system adds them all in both AVP and the signaling. If the SBC does not have a customized or default mapping for a SIP error code, it maps that code to the q850CauseTypeInternalError (47).

In addition to reason header propagation, the SBC generates reason headers based on issues it identifies itself. These would usually be 6xx responses. For these situations, the SBC also includes the reason-header AVP for CDRs and SIP error messages for egress. The system uses the default or customized messaging configured in the **local-response-map** for its own reason headers.

Note

The SBC never uses SIP cause and text to construct its own reason headers. Instead, it adds the q850 cause and text. It does, however, make the content of Reason-Header AVP the same as that of received SIP Reason: header.

To create reason headers, the SBC concatenates the q.850-cause and q.850-reason. A formatted example of a SIP reason-header follows.

```
Q.850;cause=16;text="Call Terminated"
```

See [Configure Reason and Cause Mapping for SIP-SIP Calls](#) for instructions on configuring a **local-response-map**, including the **local-error** parameters and the **add-reason-header** parameter.

Default Reason Header Information

When you enable the **add-reason-header** in the **sip-config** without having configured a **local-response-map**, the SBC adds reason information to responses and CDRs using default q.850 reason-text and q.850 cause-codes. An example of default text for the 488 message code is "Not Acceptable here". The SBC includes all error code enumerations in the local-error listing within the **local-response-map**. Default values are presented below.

| SIP Error Codes | Integer Value | Q850 Cause Codes | Integer Value |
|---------------------------|---------------|----------------------------------|---------------|
| RESP_STATUS_MOVE D | 301 | q850CauseTypeNumber Changed | 22 |
| RESP_STATUS_BAD | 400 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_UNAUT H | 401 | q850CauseTypeIncomin gBarred | 55 |
| RESP_STATUS_PAY_R EQ | 402 | q850CauseTypeIncomin gBarred | 55 |
| RESP_STATUS_FORBI DDEN | 403 | q850CauseTypeIncomin gBarred | 55 |
| RESP_STATUS_NOT_F OUND | 404 | q850CauseTypeDestUnr eachable | 3 |
| RESP_STATUS_NOT_A LLOW | 405 | q850CauseTypeIncomin gBarred | 55 |

| SIP Error Codes | Integer Value | Q850 Cause Codes | Integer Value |
|-----------------------------|---------------|---------------------------------------|---------------|
| RESP_STATUS_NOT_A CCEPT | 406 | q850CauseTypeIncomin gBarred | 55 |
| RESP_STATUS_AUTH_ REQ | 407 | q850CauseTypeIncomin gBarred | 55 |
| RESP_STATUS_REQ_T MO | 408 | q850CauseTypeNoUser Responding | 18 |
| RESP_STATUS_CONFL ICT | 409 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_GONE | 410 | q850CauseTypeNumber Changed | 22 |
| RESP_STATUS_LEN_R EQ | 411 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_TOO_B IG | 413 | q850CauseTypeInvalidM essageRecv | 95 |
| RESP_STATUS_URI_T OO_BIG | 414 | q850CauseTypeInvalidM essageRecv | 95 |
| RESP_STATUS_MEDIA | 415 | q850CauseTypeMediaN egFailure | 65 |
| RESP_STATUS_BAD_E XT | 420 | q850CauseTypeInvalidM essageRecv | 95 |
| RESP_STATUS_TMP_U NAVAIL | 480 | q850CauseTypeNouserA nswer | 19 |
| RESP_STATUS_NO_EX IST | 481 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_LOOP | 482 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_TOOM NY_HOPS | 483 | q850CauseTypeExchang eRoutingError | 25 |
| RESP_STATUS_ADDR_ INCMPL | 484 | q850CauseTypeInvalidN umberFormat | 28 |
| RESP_STATUS_AMBIG UOUS | 485 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_BUSY_ HERE | 486 | q850CauseTypeUserBus y | 17 |
| RESP_STATUS_CANCE LLED | 487 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_NOT_H ERE | 488 | q850CauseTypeMediaN egFailure | 65 |
| RESP_STATUS_INT_ER R | 500 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_NOT_I MPL | 501 | q850CauseTypeFacilityR ejected | 29 |
| RESP_STATUS_BAD_G TWY | 502 | q850CauseTypeDestOut ofOrder | 27 |
| RESP_STATUS_SVC_U NAVAIL | 503 | q850CauseTypeInternal Error | 47 |
| RESP_STATUS_GTWY_ TMO | 504 | q850CauseTypeTimeout | 102 |
| RESP_STATUS_BAD_V ER | 505 | q850CauseTypeInvalidM essageRecv | 95 |
| RESP_STATUS_BUSY | 600 | q850CauseTypeUserBus y | 17 |

| SIP Error Codes | Integer Value | Q850 Cause Codes | Integer Value |
|----------------------------|---------------|----------------------------------|---------------|
| RESP_STATUS_DECLI NE | 603 | q850CauseTypeCallReje cted | 21 |
| RESP_STATUS_DONT_ EXIST | 604 | q850CauseTypeDestUnr eachable | 3 |
| RESP_STATUS_NOTAC CEPT | 606 | q850CauseTypeInternal Error | 47 |

Changing Default Reason Text

You configure the Q850 cause code using the **entries** configuration in the **local-response-map** table for SIP Error Codes. You configure Q850 cause codes corresponding to each local error one at a time. The example configuration below sets the Q850 cause code and reason text corresponding to the locally generated SIP error 488.

```
ORACLE# sh running-config local-response-map
local-response-map
entries
local-error      not-acceptable-here-488
sip-status       488
q850-cause       65
sip-reason
q850-reason      Not Acceptable Here
```

Assume the SBC generates a 488. Using the **entries** configuration above, to specify the Q850 cause code. The SBC would construct the SIP Reason Header using the q850-cause and q850-reason.

The resulting message would be:

```
Reason: Q.850;cause=65;text="Not Acceptable Here"
```

Similarly, for 5xx, 6xx as per the Q850 cause code and reason text configured in local-response-map, SBC constructs the reason header using these values.

```
ORACLE# sh running-config local-response-map
local-response-map
entries
local-error      sip-locally-generated-bye
sip-status
q850-cause       16
sip-reason
q850-reason      Bye Call Terminated
```

The resulting message would be:

```
Reason: Q.850;cause=16;text="Bye Call Terminated"
```

If you do not configure the message, the default resulting message would be:

```
Reason: Q.850;cause=16;text="Call Terminated"
```

Configure the System to Add Reason Headers

To enable the SBC to add the Reason header:

1. In Superuser mode, type `configure terminal` and press Enter.

```
ORACLE# configure terminal
```

2. Type `session-router` and press Enter.

```
ORACLE(configure)# session-router
```

3. Type `sip-config` and press Enter.

```
ORACLE(session-router)# sip-config  
ORACLE(sip-config)#
```

4. `add-reason-header`—Enable this parameter to add the Reason header(s) to responses and CDRs.

The default value is disabled. The valid values are:

- enabled | disabled

5. Save and activate your configuration for changes to take effect.

RADIUS CDR Content Control

CDRs can be customized to limit their size as generated. The SBC's RADIUS accounting provides a detailed set of records that can contain, for example, multiple media flow descriptions for forked calls that can contain multiple sets of media and QoS attributes. While the level of detail might be required for some networks, in others the large CDRs generated to reflect that level of granularity can cause issues for the application receiving the records.

You can control the size of the RADIUS CDRs your SBC produces:

- Duplicate RADIUS attribute prevention—Using this feature, you can configure the SBC to send only one set of RADIUS attributes in CDR for a forked call. (When a forked SIP INVITE contains media information, media and QoS attributes can potentially be duplicated.)
- RADIUS attribute selection—You can set a list of the Oracle VSAs you want included in a RADIUS CDR, and the SBC will exclude the others from the record; standard attributes are always included. You specify attributes using their unique identifier in a comma-delimited list, and you can list them in any order. However, entering an invalid range disables this feature.

The SBC excludes attributes from the records in which they are already defined. If an attributes only appears in a Stop record, then it will be deleted from Stop records.

Configuring Duplicate RADIUS Attribute Generation

To enable duplicate RADIUS attribute prevention:

1. Access the **account-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

2. Type **select** to begin configuring this object.
3. **prevent-duplicate-attrs**—Enable this parameter to prevent the SBC from duplicating attributes in the accounting records it generates. This duplication can be caused, for example, by multiple media sessions within the context of a call. Retaining the default (disabled) allows the SBC to include duplicate attributes in RADIUS, Diameter and Local accounting records. This can result in attribute placement and counts that are less consistent.
4. Save and activate your configuration.

Configuring Specific RADIUS Attributes for CDR Inclusion

You enter the list of VSAs that you want included as a comma-delimited list. There are special entry types you can use in the comma-delimited list to set ranges and make entries easier:

- X- — Where X is a VSA identifier, the SBC will include all attributes with an identifier equal to or greater than X.
- -X — Where X is a VSA identifier, the SBC will include all attributes with an identifier equal to or less than X.
- - — Use the minus sign (-) alone when you want to turn off attribute selection, including all VSAs in the CDR.

To enter a list of RADIUS attributes to include in a CDR:

1. Access the **account-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

2. Type **select** to begin configuring this object.
3. **vsa-id-range**—Enter a comma-delimited list that represents the VSA you want to appear in the RADIUS CDR. There is no default for this parameter.

Do not use <Spaces> when typing in your comma-delimited list.

```
ORACLE(account-config)# vsa-id-range -5,7,10-
```

This entry specifies that CDRs contain VSA with identifiers equal to and less than 5, VSA 7, and VSAs with identifiers equal to and greater than 10.

Limit this list to accounting VSAs. For example, VSA 254 is an authentication VSA, so it should not be included in the range. The system generates validate-config errors if your range includes VSAs that are not accounting VSAs.

4. Save and activate your configuration.

Custom RADIUS CDR VSAs for SIP

This section describes how you can customize your call detail records (CDRs) for RADIUS accounting:

- Generating CDRs with call detail information from a SIP message—The SBC reserves a set of vendor-specific attributes (VSAs) and then populates them according to your header manipulation (HMR) configuration
- Generating CDRs with trunk group information—You can enable your SBC to provide terminating trunk-group and trunk-context data even when the SBC is not performing trunk-group routing.

Both support using the CSV file for RADIUS records, which you can either save locally or push to a defined FTP server.

About User-Defined VSAs for SIP Calls

The SBC reserves VSAs 200 to 230 for you to define for use with SIP calls. These VSAs should never be used for other purposes.

Because RADIUS START records are created on session initiation, their content cannot be updated. However, the content for INTERIM and STOP records can be.

To configure user-defined VSAs for a SIP call, you use HMR. For example, you could define rules so that the SBC stores the originating or terminating country code from the SIP username field in a SIP header named **P-Acme-VSA**. When generating the CDR, the SBC finds the **P-Acme-VSA** header, generates a VSA for the country code, and then includes that VSA in the CDR for the call.

You can include multiple custom VSAs in the CDR by adding the corresponding number of rules to add the header as many times as required.

Note

The same HMR configuration applies to both RADIUS and Diameter accounting, whichever is in use. AVPs 123 to 153 correspond to VSAs 200 to 230 and are reserved for HMR use.

See *Header Manipulation Rules Guide* for detailed information about HMR, including functionality, configuration, and examples.

HMR Adaptations

The following HMR element rule types support user-defined VSA for SIP calls:

- **uri-user-only**—The **uri-user-only** element rule type represents the URI username without the URI user parameters. You can perform these actions for the **uri-user-only** type: store, replaces, delete, and add. This means, for example, that you can add a username string to SIP or TEL URI without having any impact on other parameters.

- **uri-phone-number-only**—The **uri-phone-number-only** applies when all rules are met. It refers to the user part of the SIP/TEL URI without the user parameters when the user qualifies for the BNF shown here:

```

uri-phone-number-only = [+]1*(phone-digit / dtmf-digit / pause-
character)
phone-digit           = DIGIT / visual-separator
DIGIT                 = "0" / "1" / "2" / "3" / "4" / "5" / "6" /
"7" / "8" / "9"
visual-separator      = "-" / "." / "(" / ")"
dtmf-digit            = "*" / "#" / "A" / "B" / "C" / "D"
pause-character       = "p" / "w"

```

Once the URI user part qualifies as a **uri-phone-number-only** based on this BNF, the SBC ignores the visual separators when comparing it against a match value. Furthermore, the SBC performs on or using the **uri-phone-number-only** after the excluding the visual separators.

But anew value being added as a **uri-phone-number-only** or replacing a **uri-phone-number-only** does not have to match the BNF noted above. That is, you can use the **uri-phone-number-only** type knowing that:

- The action only occurs if the URI username matches the BNF defined here.
- Even so, you can also replace the **uri-phone-number-only** with one that does not match—using the same rule.

HMR String Variable

HMR supports the use of a string variable that you can use to populate headers and elements. You set this value in the **manipulation-string** parameter for a realm, SIP session agent, or SIP interface. Then, you reference it as the \$MANIP_STRING variable.

When a message arrives, the SBC matches the string you provision to the closest session agent, realm, or SIP interface. The precedence for matching is in this order: session agent, realm, and then SIP interface. For example, the SBC populates messages matching a session agent using the \$MANIP_STRING variable, but it leaves the value empty for session agents that do not match.

You can use the string variable, for instance, for values specific to realms and session agents such as country code values when the regular expression pattern used to match a country code fails to do so.

Configure User-Defined VSAs

This section shows you how to configure user-defined VSAs for SIP calls. It also contains subsections with configuration examples so you can see how this feature is put to use.

To create a header manipulation rule that generates user-defined VSAs for SIP calls:

1. Access the **sip-header-rules** configuration element.

```

ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-manipulation
ORACLE(sip-manipulation)# header-rules
ORACLE(sip-header-rules)#

```

2. **name**—Enter a meaningful name for the header rule you are creating. For example, if you want to add VSA 200 to your CDRs (or AVP ID 123, for Diameter accounting) for SIP calls, you might name your rule **generateVSA200**. There is no default for this parameter, and it is required.
3. **header-name**—Set this parameter to **P-Acme-VSA** so the SBC will add this accounting information to CDRs for the call.
4. **action**—Set this parameter to **add**.
5. **new-value**—Enter the regular expression value for the new value you want to add. For example, to add VSA 200 (or AVP ID 123, for Diameter accounting) that contains the value from the SIP From header, you would enter **200:+\$storeFrom.\$0**.

Note

The same configuration applies to VSAs for RADIUS accounting and AVPs for Diameter accounting. For both types, use the VSA number from 200-229. These correspond to AVP IDs 123-153.

6. Save and activate your configuration.
7. Add the **sip-manipulation** ruleset name to the **in-manipulationid** parameter for the relevant **real-config**, **sip-interface**, or **session-agent**.

The first example shows you how to generate custom VSA for the **to** and **from** headers in SIP messages using the following rules:

- The **storeFrom** rule tells the SBC to store the value of the **from** header for INVITES in the **\$storeFrom** variable.
- The **storeTo** rule tells the SBC to store the value of the **to** header for INVITES in the **\$storeTo** variable.
- The **generateVSA200** header rule tells the SBC to add the **P-Acme-VSA** header with value from the **storeFrom** rule. When generating the CDR, the SBC includes VSA 200 (equivalent to AVP ID 123 for Diameter accounting) with this value.
- The **generateVSA220** rule tells the SBC to add the **P-Acme-VSA** header with value from the **storeTo** rule. When generating the CDR, the SBC includes VSA 220 (equivalent to AVP ID 143 for Diameter accounting) with this value.

```

sip-manipulation
  name          customVSA1
  description
  header-rule
    name        storeFrom
    header-name from
    action       store
    comparison-type pattern-rule
    match-value .*
    msg-type     request
    new-value
    methods      INVITE
  header-rule
    name        storeTo
    header-name to
    action       store

```

```

        comparison-type      pattern-rule
        match-value          .*
        msg-type             request
        new-value
        methods              INVITE
header-rule
        name                 generateVSA200
        header-name          P-Acme-VSA
        action               add
        comparison-type      case-sensitive
        match-value
        msg-type             any
        new-value            200:+$storeFrom.$0
        methods              INVITE
header-rule
        name                 generateVSA220
        header-name          P-Acme-VSA
        action               add
        comparison-type      case-sensitive
        match-value
        msg-type             any
        new-value            220:+$storeTo.$0
        methods              INVITE

```

The second example shows you how to configure HMR to generate VSA 225 (or AVP ID 148 for Diameter accounting), which contains the customer **P_From** header when it is present. When that header is not present, the rule instructs the SBC to include the header value from the SIP **from** header for VSA 225 (AVP ID 148).

```

sip-manipulation
  name                customVSA1
  description
  header-rule
    name              storePfrom
    header-name       P_From
    action            store
    comparison-type   pattern-rule
    match-value       .*
    msg-type          request
    new-value
    methods           INVITE
  header-rule
    name              storeFrom
    header-name       from
    action            store
    comparison-type   pattern-rule
    match-value       .*
    msg-type          request
    new-value
    methods           INVITE
  header-rule
    name              generateVSA225_1
    header-name       P-Acme-VSA
    action            add
    comparison-type   case-sensitive

```

```

        match-value
        msg-type                request
        new-value                225:+$storeFrom.$0
        methods                  INVITE
header-rule
    name                        generateVSA225_2
    header-name                  P-Acme-VSA
    action                       manipulate
    comparison-type              pattern-rule
    match-value                  $storePfrom
    msg-type                     request
    new-value
    methods                      INVITE
element-rule
    name                          one
    parameter-name
    type                          header-value
    action                        delete-element
    match-val-type                 any
    comparison-type                pattern-rule
    match-value                    ^225.*
    new-value
element-rule
    name                          two
    parameter-name
    type                          header-value
    action                         add
    match-val-type                 any
    comparison-type                case-sensitive
    match-value
    new-value                      225:+$storePfrom.$0

```

ACLI Instructions String Variable

To use the HMR string variable, you set the **manipulation-string** value in the SIP session agent, realm, or SIP interface where you want the feature applied. The following sample shows you how to configure the **manipulation-string** parameter for SIP session agent.

1. Access the **session-agent** configuration element.

```

ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# session-agent
ORACLE(session-agent)

```

2. **manipulation-string**—Enter a value that references the \$MANIP_STRING variable that will be used to populate SIP headers and elements using HMR. There is no default value for this parameter.
3. Save and activate your configuration.

Trunk-Group VSA Generation

You can force the SBC to generate VSAs related to trunk groups even when you are not using the trunk group feature. With the **force-report-trunk-info** parameter turned on in the session router configuration:

- The SBC reports terminating trunk group and trunk-context information even though it has not perform trunk-group routing. The appropriate VSAs report the terminating trunk-group (VSA 65) and trunk context (VSA 67) with the information of the matching ingress session agent and realm of the originator.
- The SBC reports the terminating trunk-group (VSA 66) and trunk context (VSA 68) as the received trunk group and context from the call's SIP REQUEST message. If the SIP message has none, then the SBC uses the information from the matching egress session agent (or egress realm, when available) and next-hop realm. Note that information is reported after HMR processing—meaning that header manipulation has been performed on the message information reported.

ACLI Instructions and Examples

You enable trunk-group VSA generation on a system-wide basis in the session-router configuration.

To enable forced trunk-group VSA generation:

1. Access the **session-agent** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# session-agent
ORACLE(session-agent)
```

2. **forced-report-trunk-info**—Change this parameter to enabled if you want to turn on the SBC's ability to generate VSAs for trunk group information even when you are not using trunk-group routing. The SBC uses VSAs 65-68 to report originating and terminating trunk group information as described in the [Trunk-Group VSA Generation](#) section above. By default, this parameter is **disabled**.
3. Save and activate your configuration.

3

RADIUS Accounting Management

This chapter provides information about management and monitoring of RADIUS accounting functions on your Oracle Communications Session Border Controller including.

- alarm generation and monitoring
- status and statistics monitoring

Alarm Generation and Monitoring

The products generate alarms when certain hardware and software events occur. For more information about SBC alarms for RADIUS, refer to the Maintenance and Troubleshooting Guide.

The RADIUS ACCOUNTING CONNECTION DOWN alarm, detailed in the table below, is directly associated with the SBC's RADIUS functionality. When enabled connections to RADIUS servers have timed-out without a response from the RADIUS server, the alarm is activated. The RADIUS ACCOUNTING CONNECTION DOWN alarm triggers a Simple Network Management Protocol (SNMP) trap that is sent via the syslog Management Information Base (MIB) (ap-syslog.mib). For a list of all SNMP-related alarms and their associated traps, refer to the table of SNMP trap correlation to SBC's alarms in Oracle's MIB Reference Guide.

This alarm has no impact on a the health score of a SBC that is part of an HA Node.

RADIUS Alarms

The table below describes the SBC's alarms for RADIUS.

| Alarm | Alarm ID | Alarm Severity | Cause | Log Message | Actions |
|-----------------------------------|----------|--|---|--|---|
| RADIUS ACCOUNTING CONNECTION DOWN | 327681 | CRITICAL if all enabled and configured RADIUS accounting server connections have timed-out without a response from the RADIUS server. MAJOR if some, but not all configured RADIUS accounting server connections have timed-out without a response from the RADIUS server. | The enabled connections to RADIUS servers have timed-out without a response from the RADIUS server. | CRITICAL: All enabled accounting connections have been lost. Check accounting status for more details. MAJOR: One or more enabled accounting connections have been lost. Check accounting status for more details. | apSyslogMessage Generated trap generated critical, major dry contact syslog |

Status and Statistics Monitoring

The CLI **show radius** command, used with the three arguments described below, displays the status of any established RADIUS accounting connections and authentications. A working RADIUS connection displays READY, and a disabled connection displays DISABLED.

When an accounting server is disabled, the triggering and clearing of RADIUS ACCOUNTING CONNECTION DOWN alarms is not affected.

For more information about SBC about monitoring your SBC, refer to the Maintenance and Troubleshooting Guide.

CLI Show RADIUS Display

The **show radius** command can take one of the three available arguments:

- authentication—Shows authentication statistics for primary and secondary RADIUS servers, including: server IP address and port; round trip time; information about failed and successful requests/authentications; number of rejections; number of challenges; number of time-outs, number of retransmissions
- accounting—Shows the information described in this table:
 - Client Display—General accounting setup (as established in the accounting configuration element), including information about the state of the RADIUS client, accounting strategy used (Hunt, Failover, RoundRobin, FastestRTT, or FewestPending), IP address and port on which the server is listening, maximum message delay in seconds, and number of configured accounting servers.
 - Waiting Queue—Amount of accounting (RADIUS) messages waiting to be sent. Waiting queue capacity is 4,096 messages.
 - <IP Address:Port>—Information about each configured accounting server (established in the accounting servers configuration). The heading above each accounting server section is the IPv4 address and port combination of the accounting server described. This section also includes information about the accounting server's state (e.g., Connect_Attempt, INIT).
- all—Shows all of the information for both the authentication and accounting displays

The following is an example of the CLI **show radius authentication** command output.

```
ORACLE# show radius authentication
Active Primary Authentication Servers:
  server ipAddr: 172.30.0.7
Active Secondary Authentication Servers:
  server ipAddr: 172.30.0.8
Authentication Statistics:
  Server: "172.30.0.7:1812"
    RoundTripTime           :0
    MalformedAccessResponse:0
    AccessRequests          :2
    BadAuthenticators       :0
    AccessRetransmissions   :5
    AccessAccepts           :0
    Timeouts                :6
    AccessRejects           :0
    UnknownPDUTypes        :0
```

```

AccessChallenges      :0
Server:"172.30.0.8:1812"
    RoundTripTime      :0
    MalformedAccessResponse:0
    AccessRequests     :2
    BadAuthenticators  :0
    AccessRetransmissions :9
    AccessAccepts     :0
    Timeouts          :10
    AccessRejects     :0
    UnknownPDUTypes   :0
    AccessChallenges   :0

```

The following is an example of the ACLI **show radius accounting** command output.

ORACLE# show radius accounting

```

*****Client Display Start*****
Client State = READY, strategy=Hunt
listening on 127.0.0.1:1813
max message delay = 60 s, # of servers = 2
===== Waiting Queue =====
Waiting size = 89
=====
----- 10.0.0.189:1813 -----
Remote = 10.0.0.189:1813, Local = 0.0.0.0:1026, sock=45 (BOUND)
conn state=READY, RTT=250 ms
Min Rtt=250 ms, Max inactivity=60 s, expires at Nov 21 13:50:19.582, Restart
delay=30 s
----- 192.168.200.70:5050 -----
Remote = 192.168.200.70:5050, Local = 0.0.0.0:1027, sock=46 (BOUND)
conn state=DISABLED, RTT=0 ms
Min Rtt=250 ms, Max inactivity=60 s, expires at Nov 21 13:50:19.569, Restart
delay=30 s
*****Client Display End*****

```

The following is an example of the ACLI **show radius all** command output.

```

ORACLE# show radius all
*****Client Display Start*****
Client State = READY, strategy=Hunt
listening on 127.0.0.1:1813
max message delay = 60 s, # of servers = 2
===== Waiting Queue =====
Waiting size = 89
=====
----- 10.0.0.189:1813 -----
Remote = 10.0.0.189:1813, Local = 0.0.0.0:1026, sock=45 (BOUND)
conn state=READY, RTT=250 ms
Min Rtt=250 ms, Max inactivity=60 s, expires at Nov 21 13:50:19.582, Restart
delay=30 s
----- 192.168.200.70:5050 -----
Remote = 192.168.200.70:5050, Local = 0.0.0.0:1027, sock=46 (BOUND)
conn state=DISABLED, RTT=0 ms
Min Rtt=250 ms, Max inactivity=60 s, expires at Nov 21 13:50:19.569, Restart

```

```
delay=30 s
*****Client Display End*****
Active Primary Authentication Servers:
  server ipAddr: 172.30.0.7
Active Secondary Authentication Servers:
  server ipAddr: 172.30.0.8
Authentication Statistics:
  Server: "172.30.0.7:1812"
    RoundTripTime           :0
    MalformedAccessResponse:0
    AccessRequests          :2
    BadAuthenticators       :0
    AccessRetransmissions   :5
    AccessAccepts           :0
    Timeouts                 :6
    AccessRejects           :0
    UnknownPDUTypes        :0
  AccessChallenges          :0
  Server: "172.30.0.8:1812"
    RoundTripTime           :0
    MalformedAccessResponse:0
    AccessRequests          :2
    BadAuthenticators       :0
    AccessRetransmissions   :9
    AccessAccepts           :0
    Timeouts                 :10
    AccessRejects           :0
    UnknownPDUTypes        :0
    AccessChallenges        :0
```

4

Local CDR Files and Push Receivers

The local CDR file feature allows you to save RADIUS CDR data to a local CSV text file on local drive on the SBC. Local CDR file creation and storage can be used in addition to or independently of sending CDRs to RADIUS servers for calls. With the local CDR files, you can:

- Send the files to a remote server by configuring a push receiver
- Develop your own scripts for retrieving the files from the SBC

You configure the SBC to:

- Set directory path where you want to save local CDR files
- Set a maximum file size for the CSV file
- Set a maximum number of local CDR files
- Set an interval in which to close the existing local CDR file and begin writing a new file.

Once local CDR file creation is enabled, you can configure push receivers to send closed local CDR files to a server using FTP or SFTP protocols. Once sent off-box, the SBC deletes the local CDR file that was successfully sent.

Local CDR File Format

The CDRs are written as comma-delimited ASCII records to files on the SBC. The types of records are controlled by the same accounting configuration parameters used for RADIUS. The fields of the comma-delimited entries correspond to RADIUS START, INTERIM, and STOP records. Using the accounting configuration, you can configure the SBC to record STOP records only.

Because the record types do not have consistent field positioning, any server parsing them would need to read the first field to determine the type and learn how to parse the remaining fields.

Local CDR File Format Consistency

Unpopulated or unused fields in the RADIUS CDR are omitted from the locally-stored CSV file. This means that there is no fixed position for a RADIUS attribute across all CSV files. Instead, the missing values are skipped in the CSV file so that the order and appearance for attribute values can differ from record to record.

You can guarantee the placement of attributes in local CDR files with the **cdr-output-inclusive** parameter. With this enhancement enabled, RADIUS records sent to a RADIUS client contain even empty attributes with an integer, date and time, or IP address format; the default value is zero. In other words, when there is no value to report:

- An IP address attribute will report as 0.0.0.0
- A date and time attribute will report as 00:00:00.000 UTC Jan 01 1970
- An integer attribute value will report as 0

To maintain RFC 2865 and 2866 compliance, the Oracle Communications Session Border Controller will not send empty attributes that are string values to a RADIUS client. And when you enable this feature, the Oracle Communications Session Border Controller adds all attributes to the local CDR file.

Masking local CDR fields

The content in local CDR files can also be controlled by using the [RADIUS Attribute Selection](#) feature. This lets you exclude call data fields from local CDR files.

Decoding Local CDR File Contents

The Local CDR File contents do not follow a standard CSV file format.

- Fields in that are strings are enclosed in double quotes ("). Each string may contain commas, double quotes, and other similar characters that are not escaped.
- When you parse a CDR entry, the end will be indicated by a comma character after an even number of double quotes.

```
1, "<sip:0322911111@10.200.200.30:5066>;reason="unconditional", <sip:0322917676@10.200.200.30:5066>;reason="alias", 0, 0, 112
```

Generate Local CDR Layout Files

Numerous factors determine the layout of local CDR files. In order to obtain an accurate local CDR layout, the SBC can write a special CDR layout file that only includes the data layout for your local CDRs based on your configuration. You can then use this file to interpret local CDR files with the proper data field order, source and identification label.

You can configure the system to produce CDR layout files with the **dump_csv_format** command at the superuser prompt.

```
ORACLE# dump_csv_format
```

This function uses the same process, input and output mechanisms the system uses to produce local CDRs. While this command is activated, the system produces layout files instead of actual CDRs. After the layout files have been created, turn the generation feature off with the **no_dump_csv_format** command.

Format files are written to the same directory as local CDR files, and they use the same naming convention as local CDR files. Refer to local CDR generation instructions to identify the files you intend to retrieve, based on your configuration for rotation, naming, file size, and so forth.

Perform this procedure in a maintenance window. Limit your sample calls to a single successful, and depending on your configuration, single unsuccessful call. The following is the general procedure used to capture local CDR layout files.

1. Turn on **dump_csv_format** from the system's enable prompt. The system stops generating local CDR files, generating local CDR format files instead.
2. Place a successful call.
3. Complete the call.
4. If you are configured for INTERIM generation upon an unsuccessful call, place an unsuccessful call.

5. Depending on your configuration, identify the file that has the format. For example, if using rotation you may decide to wait for the data to rotate from the temp file to be sure the file is closed.
6. Retrieve the layout file from the local CDR directory.
7. Turn off the feature using `no_dump_csv_format`. The system begins to generate local CDR files again.
8. Use the files to identify your CDR format and establish your collection and collation process.

Local CDR Layout File Reference and Example

The first line of every record contains the following comma-delimited information:

```
"1", "Accounting Status", , "40", ["## START ##" | "## INTERIM ##" | "##STOP##"]
```

Each line after the initial line of each record contains the following comma-delimited information:

```
<CDR Attribute position>,<CDR Attribute Name>,<VSA Vendor>,<VSA Number>
```

The CDR Attribute name only presents the shorthand of the attribute. Cross-reference the VSA number with the RADIUS dictionary to obtain the full VSA name.

The following is an example of the first 10 rows of a CDR Layout file, start record.

```
1,"Accounting Status",,40,## START ##
2,"NAS IP Address",,4
3,"NAS Port",,5
4,"Accounting Session ID",,44
5,"Ingress Session ID",ACME,3
6,"Egress Session ID",ACME,4
7,"Session Protocol Type",ACME,43
8,"Session-Forked-Call-Id",ACME,171
9,"Generic ID",ACME,40
10,"Calling Station ID",,31
```

Requirements

If you want to guarantee the CSV placement for RADIUS attribute values, you must use the entire RADIUS dictionary. You cannot use the RADIUS CDR abbreviation feature. Using an abbreviated form of the RADIUS dictionary results in adverse effects for the CSV file.

In your configuration, then, you must set the **vsa-id-range** parameter to use the entire range of attributes. Leaving this parameter blank disables abbreviation and all attributes are included. Alternatively, you can specify all of the parameters (by attribute number) that are used in the OS release loaded on your system.

See the [RADIUS CDR Content Control](#) section for more information.

Local CDR File Naming Convention

The SBC creates filenames from the timestamp that the CDR file is opened for writing. The format is `cdrYYYYMMDDHHMM[a-j]`, where:

- YYYY=the year
- MM=the month
- DD=the day
- HH=the hour
- MM=the minute
- [a-j]=a suffix that provides additional discrimination in case of changing system time, setting the rotation time for this feature to one minute, or in case of another occurrence that might compromise the date and time

Your file name will resemble the following sample: `cdr200511151200`.

Temporary File Naming for an Open CDR File

Before this release was introduced, the SBC used the same naming format for all CDR files: **cdrYYYYMMDDHHMM[a-j]**. If this is the naming convention you still want to use, you can do so simply by This mode offers no means of differentiating a file to which the SBC is writing information from any other closed file(s).

Open local CDR files have a **temp-** prefix added to the file name. The prefix helps you differentiate the file that is currently open from the other CDR files. As soon as the file is closed during rotation, the **temp-** prefix is removed.

You can revert to legacy behavior that omits the **temp-** prefix on open files by adding the **disable-temp-file** option to your accounting configuration.

```
ORACLE(account-config)# options +disable-temp-file
```

Operational Details

This section offers details of SBC operations that effect temporary CDR file naming.

- **Reboot**—A system reboot can happen unexpectedly, or might be caused by your intentionally rebooting the system using the ACLI **reboot** command. When a reboot occurs, SBC closes the CDR file that was most recently opened (before the reboot) and names it according to the **cdrYYYYMMDDHHMM[a-j]** convention. It also opens a new file, which bears the **temp-** differentiation.
- **Activating a configuration**—If temporary CDR naming is enabled before and after you use the **activate-config** command, then the last opened file will be closed and have the **cdrYYYYMMDDHHMM[a-j]** name format. The SBC also opens a new file with the **temp-** prefix to which it will write data.
In the case where temporary CDR naming is enabled before you activate a configuration and disabled after it, the last open file is named according to the **cdrYYYYMMDDHHMM[a-j]** name format. The new file to which the SBC will write data is also in the **cdrYYYYMMDDHHMM[a-j]** name format. In other words, the SBC does not use the **temp-** prefix designation at all.
In the case where temporary CDR naming is disabled before you activate a configuration and enabled after it, the SBC closes the most recently opened file—which must have been in the **cdrYYYYMMDDHHMM[a-j]** name format. The SBC also opens a new file with the **temp-** prefix to which it will write data.
- **Changing the accounting configuration's administrative state**—When you disable the accounting configuration, the SBC renames the most recently opened file with the **temp-** prefix to the **cdrYYYYMMDDHHMM[a-j]** name format.

HA Considerations for CDR Output Redundancy

The considerations in this section describes the Oracle Communications Session Border Controller's behavior when CDR output redundancy is enabled or disabled. You set CDR output redundancy in the accounting configurations **cdr-output-redundancy** parameter.

- **Enabled**—When you enable CDR output redundancy, both the Active and Standby systems rotate files. During CDR file rotation, if either the Active or the Standby rotates a file with the **temp-** prefix, the prefix disappears and the file name appears in the **cdrYYYYMMDDHHMM[a-j]** name format.

The Active and the Standby systems always have the same files, including the CDR file with the **temp-** prefix. So the file exists on both systems.

To support dynamic configuration on SBC systems deployed as an HA pair when **cdr-output-redundancy** is enabled, the system may intermittently display the following fields under **show health** as **false** when you save and activate configuration changes.

- Rotated CDRs Synchronized
- Radius CDR Synchronized

This is an expected behavior and does not impact any CDR activities.

- **Disabled**—When you have disables CDR output redundancy and switchover happens for any reason, it is key that there are no residual files with the **temp-** prefix. For this reason, the SBC handles the situation as follows:

Becoming Active—When it transitions from Standby to Active, a SBC checks for any files with the **temp-** prefix, closes the file if it is open, and renames it according to the **cdrYYYYMMDDHHMM[a-j]** name format. These actions means that the file is not only renamed, but that it is also rotated. Rotation triggers the creation of a new CDR file with the **temp-** prefix to use for new CDR data.

Becoming Standby—When it transitions from Active to Standby, a SBC closes the open **temp-** prefix file and renames it according to the **cdrYYYYMMDDHHMM[a-j]** name format. Rotation creates a new **temp-** prefix file on the Standby, which remains empty until it transitions back to the Active state.

- **Standby push**—provides the user with a mechanism to manage CDRs in a High Availability (HA) environment that prevents CDR loss during HA events. When the user configures this setting, the standby can SFTP files to the CDR server. When the decision is made to become standby the system looks for any un-pushed CDRs that are still present and pushes those to the CDR server.

This process addresses two potential data loss issues:

- When an active node fails over, there may be temporary CDRs that have not yet been pushed to the CDR server. If the standby cannot send the record to the CDR server itself, the data is lost, resulting in the call not being charged.
- Temporary CDRs files often contain incomplete data and can be present on both the active and standby SBCs. After a failover, and when the original active becomes the active again, it would send this temporary file to the CDR server and overwrite the file that had been sent from the previously active system.

The user sets this value using the syntax below.

```
ORACLE(account-config)# cdr-output-redundancy standby-push
```

Setting this value can also enhance the clarity of CDR file names. This would also require that the applicable **account-config** have its **file-seq-number** parameter enabled. The

system then includes information within the filename to distinguish whether the file came from the active or standby node. This prevents the standby from overwriting CDRs already on the server. This naming convention adds two prefixes to the CDR filename.

```
XXXX-YYYY-CDR201610171015-000000050
```

- XXXX - The filename-prefix defined in your **push-recvier** setting
- YYYY - The host name of the SBC

The **standby-push** value requires that the applicable account-config operate over a management (wancom) interface. Configuration verification fails if the user sets this value to an **account-config** operating over a media interface.

All parameters are RTC supported.

Note

Before you upgrade from a release prior to S-CZ7.2.0 to S-CZ7.2.0 or later, you must set the **cdr-output-redundancy** parameter to **enabled** for the Standby to upgrade and sync properly. You can then change the parameter to **disabled** afterwards, if needed.

Caveats

As described above, when the system reboots for any reason or when you issue an **activate-config**, the SBC checks for CDR files with the **temp-** prefix and renames to the usual **cdrYYYYMMDDHHMM[a-j]** format.

However, if you change the accounting configuration's file-path value and subsequently the system either reboots or you activate your configuration, the SBC will be unable to check for files with the **temp-** prefix in the old file path. And so it will also be unable to rename them. The SBC checks the new path only.

Call Detail Record Sequence Number in Filename

To assist in the identification of lost Call Detail Record (CDR) files, the customer can enable the **file-seq-number** attribute to assign a sequence number to append to the file. A separate configuration element, **temp-remote-file**, allows for the prepending of the characters "tmp-" to CDR files during transfer.

Sometimes local CDR transmission failures occur due to underlying network or infrastructure issues. Customers can identify missing files through the combination of a timestamp (YYYYMMDDMM) and 9-digit unique sequence numbers (SNs) appended to the file. This behavior is enabled through the **file-seq-number** attribute. The SN will start from one at boot time. This attribute replaces the use of alpha characters (a-z) appended to the CDR file name when more than one file is created in the same minute.

Sequence Number Caveat for HA Deployments

File sequence numbers can unexpectedly reset to 1 in some circumstances when operating in an HA deployment. This can happen if the file synchronization is not complete when you perform a switchover. Before performing a switchover, check the file names on both the active and the standby to ensure they are similar.

Consider the examples below. The first codeblock shows the CDR filenames on the active. Note the sequence numbers 473 and 474.

```
ORACLE#show directory /opt/cdr
rw-rw-r- 1 root root 20 Apr 1 02:26 cdr202504010225-000000473.gz
rw-rr- 1 root root 0 Apr 1 02:26 temp-cdr202504010226-000000474
```

Having identified the active's filenames, check the filenames on the standby. In the example below, both 473 and 474 are present, with an earlier file that uses sequence number 472.

```
ORACLE#show directory /opt/cdr
rw-rw-r- 1 root root 20 Apr 1 02:25 cdr202504010223-000000472.gz
rw-rw-r- 1 root root 20 Apr 1 02:26 cdr202504010225-000000473.gz
rw-rr- 1 root root 0 Apr 1 02:26 temp-cdr202504010226-000000474
```

In this case, you can perform the switchover without having the system reset the sequence numbers.

The next codeblock shows the CDR filenames on the active. Note the sequence numbers 487 and 488.

```
ORACLE#show directory /opt/cdr
rw-rw-r- 1 root root 20 Apr 1 02:52 cdr202504010250-000000487.gz
rw-rr- 1 root root 0 Apr 1 02:52 temp-cdr202504010252-000000488
```

Having identified the active's filenames, check the filenames on the standby. In the example below, the sequence numbers on the standby are 484 and 485. In this case, file synchronization is not yet completed.

```
ORACLE#show directory /opt/cdr
rw-rwxr- 1 root root 20 Apr 1 02:46 cdr202504010244-000000484.gz
rw-rwxr- 1 root root 0 Apr 1 02:46 cdr202504010246-000000485
```

If you perform the switchover with different sequence numbers on the standby, the system resets the sequence number to 1 after the switchover, as shown below.

```
ORACLE#show directory /opt/cdr
rw-rw-r- 1 root root 20 Apr 1 02:53 cdr202504010252-000000488.gz
rw-rr- 1 root root 0 Apr 1 02:53 temp-cdr202504010253-000000001
```

To avoid this behavior, ensure that the CDR files are synchronized on the active and standby before you perform the switchover.

Local CDR File Storage Directories

The SBC only allows local storage of CDRs to the `/opt` directory. If you try to save to another directory (such as `/code` or `/boot`), you will receive an error message.

Setting the location to `/opt/logs` may cause the `package-logfiles` command and the `package-crashfiles` command to fail. Instead, use `/opt/accounting`.

If you are using the ACLI and enter an inappropriate directory, the ACLI will issue an error message.

Local CDR File Size and Rotation

You can configure maximum file size, maximum number of local CSV files to store, and the interval at which the files rotate.

The SBC saves up to the file size limit (**max file size**) and maintains only number of files that you configure (**max files**). When the maximum file size is reached, the SBC closes that file and begins writing VSA attributes and values to a new local CDR file. When it is time for the SBC to write the **max files** + 1 file, the oldest file is deleted so that the newest one can be stored.

You can set the period of time at which the SBC ends writing a local CDR file and begins writing data to a new file. This is the file rotate time and is configured in the **file-rotate-time** parameter.

Local CDR File Compression

You can configure the SBC to compress local CDRs in zip format to save disk space by setting the **file-compression** parameter to enabled.

Local CDR File Redundancy

The SBC can create a redundant copy of the local CDR files, and store them on the standby system in the HA node.

This enhancement to the CDR storage feature ensures against data loss if, for example, an active SBC fails immediately before sending the files off-box to a push receiver. The standby has a duplicate set of records that it sends. This feature is enabled with the **CDR output redundancy** parameter found in the **account config** configuration element.

Caveats for H.323

H.323 calls proceed without interruption over an HA node in the event of a failover from one SBC to another, and RADIUS records are generated and duplicated across the active and standby systems in an HA node. However if a switchover occurs during an H.323 call (that has been initiated, but not completed), the newly active (formerly standby) system will not generate RADIUS Stop records when the call completes.

Configuring Local CDR Files

This procedure explains how to enable and customize creation of local CDR files.

1. Access the **account-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

2. **file-output**—Enable this parameter for the SBC to create comma-delimited CDRs (generated from RADIUS records). By default, this parameter is disabled.
3. **file-path**—You must configure this path for the CDR push feature to work. Set the path to use on the SBC for file storage:

- /opt
- /opt/logs

To use FTP push, you must configure a usable path.

Note

When a Hard Disk Drive is available, you may opt to store CDRs in the data-disk.

4. **max-file-size**—Set the maximum CDR file size in bytes. The default and minimum value is 1000000. Oracle recommends you limit local CDR storage on your system to 30M. For example, if you retain the **max-file-size** default, set **max-files** to 30. However, if you are using a Storage Expansion Module the maximum value is 108.
5. **max-files**—Set the maximum number of files to be stored on the SBC at one time. The parameter's value range is from 0 to unlimited. The user should consider the max-file-size setting, the 30M recommendation, and their preferences to specify this value. The default is 5.
6. **file-sequence-number**—set this to enabled for the system to assign a 9 digit file sequence number to append to a CDR file. The default is disabled.
 - **enabled | disabled**
7. **file-compression**—set this to enabled for the system to compress closed local CDR files in .zip format.
8. **file-rotate-time**—Set how often in minutes you want to rotate the stored files; the SBC overwrites the oldest file first. The minimum rotation time is 0 minutes, the default. Leaving this value set to the default means that the SBC does not rotate the files.
9. **cdr-output-redundancy**—Set this parameter to enabled for the SBC to store a redundant copy of the local CSV file to the standby HA node. Set this parameter to standby-push to enable the function and enable local storage and implement push capabilities on an HA deployment's standby to protect against CDR data loss.
10. **vsa-id-range**—Either leave this parameter blank (default), or enter the complete range of VSAs loaded on your system. The following example shows what you might enter to use all of the VSAs for a system that is not running QoS.

```
ORACLE(account-config)# vsa-id-range 1-4,10-14,20-24,28,29,32-71,74-136
```

11. **cdr-output-inclusive**—When disabled (default), the system excludes fields that have no data from the CSV file. Set to enabled to make the system include all fields in every CSV file. This ensures that there are always the same number of fields in all equivalent records. Start records would always have the same number of fields. The same would be true of interim and stop records.
12. Type **done** to save your configuration.

Local CDR File Delete Warning

You can configure the SBC to initiate an alarm and send an SNMP trap when the oldest local CDR file was deleted under fault conditions. This feature is enabled with the **file delete alarm** parameter.

The SBC deletes a local CDR file in the following three cases:

1. After the local CDR file has been successfully transferred to a push receiver
2. The number of local CDR files exceed the limit configured in the **account-config, max-files** parameter
3. No free disk space remains on the partition where the local CDR files are written: **account-config, file-path**

If a local CDR file is deleted after it was successfully uploaded to a push receiver, no fault is triggered because this is standard, expected operation. But if a local CDR file is deleted for case 2 or 3 above, it is considered a fault condition initiating an alarm and SNMP trap.

Local CDR File Delete Alarm

The CDR file delete alarm is configured in **account config** configuration element by enabling the **file-delete-alarm** parameter. This is a minor severity alarm and is non-health affecting. This alarm has no clearing condition and must be manually cleared.

Local CDR File Delete SNMP Trap

Under the same circumstances that cause a CDR file delete alarm, an SNMP trap will be sent to all configured trap-receivers. The `apSysMgmtCdrFileDeleteTrap` trap contains the following information:

- File Name—name of the file that was deleted

Local CDR Push Receivers

Local CDR push receivers are used to transfer local CDR files to a remote server on a periodic basis. You configure this feature by defining push receivers with a remote IP address and port and login credentials for FTP and SFTP servers. At the configured time interval (**file rotate time**), the SBC closes the current file, and transfers the files that are complete and have not yet been pushed; including the just-closed-file. These files are uploaded to the remote servers at a specified **remote-path**. You can choose either ftp or sftp protocols to access the remote servers.

In the case that all push receivers are unreachable, then local CDR files continue to be written to local file system until the push receivers return to service. Once a push receiver becomes reachable, the SBC transfers all local CDR files to the remote server automatically. After all local CDR files have been successfully transferred to server, they are deleted from the local volume.

You can optionally set the **temp-remote-file** attribute so the characters "tmp-" are prepended to the CDR file during transfer. Once delivered, the file will be renamed on the remote host to remove "tmp-".

Secure FTP Push Configuration

You can configure the Oracle Communications Session Border Controller (SBC) to securely log on to a push receiver using one of the following methods that creates a secure connection.

Password authentication—

1. Set the **protocol** parameter on the push receiver to SFTP.
2. Configure a username and password.
3. Leave the **public-key** parameter blank.

4. Import the host key from the SFTP server to the SBC as a known-host key. See "Manage SSH Keys" in the *Configuration Guide*.

Public key authentication—

1. Set the **protocol** parameter on the push receiver to SFTP.
2. Configure the username.
3. Leave the **public-key** parameter blank, regardless of authentication type.
4. Export the SBC's public key with the `show security public-host-key rsa` command.
5. Append the SBC's public-key to the SFTP server's `authorized_keys` file.
6. Import the host key from the SFTP server to the SBC as a known-host key. See "Manage SSH Keys" in the *Configuration Guide*.

It is often difficult to determine whether the SFTP server uses its RSA key or its DSA key for its server application. For this reason, Oracle recommends that you import both the RSA key and the DSA key to the SBC to ensure a successful FTP Push.

It is also common for the SFTP server to run the Linux operating system. For Linux, the command `ssh-keygen -e` creates the public key that you need to import to the SBC. The `ssh-keygen -e` command sequence requires you to specify the file export type, as follows.

```
[linux-vpn-1 ~]# ssh-keygen -e
Enter file in which the key is (/root/.ssh/id_rsa/): /etc/ssh/
ssh_host_rsa_key.pub
```

If you cannot access the SFTP server directly, but you can access it from another Linux host, use the `ssh-keyscan` command to get the key.

```
root@server:~$ ssh-keyscan -t rsa sftp.example.com
```

Multiple Push Receivers

SBC (SBC) supports up to five CDR push receivers for use with the local file storage and FTP push feature. For each receiver you configure, you can set the file transfer protocol that you want to use. (FTP or SFTP). The system uses the push receivers according to the priorities you assign by setting a 0 through 4 priority number to the server. 0 is the highest priority, and 4 (default) is the lowest.

Based on the priority level you set, the SBC uses the strategy that you set to select a CDR push receiver. If the highest priority push receiver selected using the strategy becomes unavailable, the SBC uses the strategy (hunt, round robin) to select another.

This feature is dynamically configurable. When you change the configuration, the SBC updates the list of push receivers if it has changed.

Deprecated ACLI Configuration

The following parameters in the `account-config` configuration element are deprecated:

- `ftp-address`
- `ftp-port`
- `ftp-user`

- ftp-password
- ftp-remote-path

These parameters will only be used if no **account-config**, **push-receiver** configuration elements have been defined. All new push receivers must be defined in the **account-config**, **push-receiver** configuration element.

Configuring Local CDR Push Receivers

This procedure explains how to enable and customize a push receiver configuration that will transfer closed local CDR files to remote servers.

1. Access the **account-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

2. **ftp-push**—Set the state of FTP push feature to **enabled**. It is disabled by default.
3. Type **push-receiver** and press Enter.

```
ORACLE(account-config)# push-receiver
```

You can now begin to configure a push receiver.

4. **server**—Enter the IP address of this push receiver.
5. **port**—Enter the port number of this push receiver.
6. **remote-path**—Enter the remote pathname where local CDR files should be written on the push receiver.
7. **filename-prefix**—Enter the filename prefix (as a string) to prepend to the Local CDR files. The SBC does not rename local files. There is no default for this parameter.
8. **priority**—Enter a number 0 through 4 to set the priority of this push receiver in relation to the others you configure on the system. The highest priority, and the push receiver the system uses first, is 0. The lowest priority, and the push receiver the system uses last, is 4 (the default).
9. **temp-remote-file**—set the state of this element to enabled for the system to prepend the characters "tmp-" to a CDR file during transfer. When the transfer ends successfully, the system removes the characters "tmp-". The default is disabled
 - **enabled | disabled**
10. **protocol**—Enter **SFTP** or **FTP**. Consider the **port** value you configured earlier.
11. **username**—Enter the username the SBC uses when connecting to this push receiver. There is no default for this parameter. This parameter is always required.
12. **password**—Enter the password corresponding to the username the SBC uses when connecting to this push receiver. There is no default for this parameter. You can leave this field blank if you are using public key authentication. Profile configuration is required for both password and public key authentication.
13. **public-key**—Do not configure this field. Public key operations are performed outside of this configuration element. See "SSH Key Management" in the *Configuration Guide*.

14. **ftp-strategy**—Set the strategy you want the SBC to use when selecting from multiple push receivers. The default is **hunt**.
 - Hunt—The SBC selects the push receiver from the available list according the priority level. The system uses this strategy as its default.
 - Failover—The SBC selects the push receiver based on priority level and will continue to use that same push receiver until it fails over.
 - RoundRobin—The SBC selects push receivers systematically one after another, balancing the load among all responsive push receivers.
 - FastestRTT—The SBC selects the push receiver based on best average throughput. For this situation, throughput is the number of bytes transferred divided by the response time. The system uses a running average of the five most recent throughput values to accommodate for network load fluctuations.
15. **ftp-max-wait-failover**—Enter the amount of time in seconds to wait before the SBC declares a push receiver to have failed over. This default value for this parameter is **60**.
16. Type **done** to save your configuration.

Monitoring CDR Push Receivers

The SBC provides several mechanisms to monitor push receivers

ACLI Show Command

You can use the ACLI **show radius cdr** command to view information about CDR push receivers. The existing display for this command has been extended to include information that looks like the following:

```
***** CDR Push Receiver Display Start*****
strategy = FastestRTT, maxwaitfailover = 10, number of receivers = 1
----- 172.30.0.70:21 -----
cdrpush-receiver = 172.30.0.70:21, state = READY, priority = 4
remote path = /home/acme, remote prefix = vik, protocol = ftp
username = acme, password = *****, publickey =
FTP rtt = 0, FTP successes = 0, FTP failures = 0
FTP timeouts = 0, FTP Delays = 0, FTP Put failures = 0
FTP conn failures = 0, FTP terminates = 0, FTP triggered terminates = 0
```

SNMP Traps and Alarms

The SBC sends traps when a single push receiver or all push receivers become unavailable.

- When one CDR push receiver becomes unavailable, the `CDR_PUSH_RECEIVER_FAIL_TRAP` trap is sent and a minor alarm is generated.
- When all of the configured CDR push receivers become unavailable, the `CDR_ALL_PUSH_RECEIVERS_FAIL_TRAP` is sent and a major alarm is generated.

When one or more of the push receivers comes back, the `CDR_ALL_PUSH_RECEIVERS_FAIL_CLEAR_TRAP` is sent and the alarm is cleared.

The SBC sends out traps and triggers corresponding alarms when it encounters failure when attempting to transfer local CDR files. One set of traps is used for single-instance push receiver failures; another set is used when all receivers fail. They are part of the `apSysMgmtCDRPushReceiverNotificationsGroup`.

All of the traps contain information about the type of push receiver, the address of the push receiver, and the failure reason code.

The trap and corresponding clearing trap for single push receiver failure are:

- apSysMgmtCDRPushReceiverFailureTrap
- apSysMgmtCDRPushReceiverFailureClearTrap

A minor alarm is generated when the apSysMgmtCDRPushReceiverFailureTrap is sent.

The trap and corresponding clearing trap for global push receiver failure are:

- apSysMgmtCDRAIIPushReceiversFailureTrap
- apSysMgmtCDRAIIPushReceiverFailuresClearTrap

A major alarm is generated when the apSysMgmtCDRAIIPushReceiversFailureTrap is sent.

5

CDR Attribute Reference

This section describes the Vendor Specific Attributes (VSA) that the Oracle Communications Session Border Controller supports.

The Oracle Communications Session Border Controller supports CDRs through RADIUS reporting with additional VSAs to include information that is not available with the standard RADIUS session information. CDRs provide billing information on sessions traversed through a system, as well as troubleshooting information, fraud detection, fault diagnostics, and service monitoring.

CDRs can contain information about recent system usage such as the identities of sources (points of origin), the identities of destinations (endpoints), the duration of each call, the amount billed for each call, the total usage time in the billing period, the total free time remaining in the billing period, and the running total charged during the billing period. VSAs are defined by vendors of remote access servers in order to customize how RADIUS works on their servers.

Standard RADIUS Attributes

This section describes the standard RADIUS attributes that the SBC supports. These attributes appear along with VSAs (Vendor-Specific Attributes) in the CDRs that the SBC generates.

The [Standard RADIUS Attributes Dictionary](#) is a dictionary of the standard RADIUS attributes included in Accounting Request messages sent by the SBC to the RADIUS server. The CDR event information determines which messages are generated and which RADIUS attributes are included in the messages. Standard RADIUS messages and attributes are used whenever possible; however, RADIUS does not have attributes to record all important session information.

Possible messages are:

- Start—Marks the start of service delivery and describes the type of service being delivered and the user to whom it is being delivered
- Interim-Update—Indicates to the accounting server that the session parameters have changed
- Stop—
 - Marks the end of service delivery
 - Describes the type of service that was delivered
 - Sometimes describes statistics such as elapsed time, input and output octets, or input and output packets
- On—Marks the start of accounting
- Off—Marks the end of accounting

VSAs are used to record the necessary session information missing from this list of standard RADIUS attributes.

For more information about RADIUS, see to the following Internet Engineering Task Force Request for Comments (IETF RFCs):

- RFC 2865, Remote Authentication Dial In User Service (RADIUS), Rigney, et al., June 2000 (<http://www.ietf.org/rfc/rfc2865.txt>)
- RFC 2866, RADIUS Accounting, C. Rigney, June 2000 (<http://www.ietf.org/rfc/rfc2866.txt>)

Standard RADIUS Attributes Dictionary

The table below lists and describes standard RADIUS attributes.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|--------------------|---|-----------------|----------------------|--|
| NAS-IP-Address | IP address of the SIP proxy or the H.323 stack's call signaling address. | 4 | IP address | <ul style="list-style-type: none"> • Start • Interim-Update • Stop • On • Off |
| NAS-Port | SIP proxy port or the H.323 stack's call signaling RAS port. | 5 | integer | <ul style="list-style-type: none"> • Start • Interim-Update • Stop • On • Off |
| Called-Station-Id | To field value of the SIP INVITE message (a type of message used to initiate a session) or the calledPartyNumber of the H.323 message. | 30 | string | <ul style="list-style-type: none"> • Start • Interim-Update • Stop |
| Calling-Station-Id | From field value of the SIP INVITE message or the callingPartyNumber of the H.323 message. | 31 | string | <ul style="list-style-type: none"> • Start • Interim-Update • Stop |
| NAS-Identifier | <p>Value, if any, set in the optional NAS-ID field for the accounting server that you configure as part of the accounting configuration. This identifier sets the value that the remote server (the accounting server) uses to identify the SBC so that RADIUS messages can be transmitted.</p> <p>The remote server to which the accounting configuration will send messages uses at least one of two pieces of information for identification:</p> <p>NAS IP address: always included in the accounting message</p> <p>NAS identifier: configured in the NAS-ID parameter of the accounting server; if configured, the NAS identifier is sent to the remote server</p> <p>This attribute only appears if a value is configured in the NAS-ID field.</p> | 32 | string | <ul style="list-style-type: none"> • Start • Interim-Update • Stop • On • Off |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|----------------------|--|-----------------|----------------------|--|
| Acct-Status-Type | Whether this Accounting Request marks the beginning of the RADIUS message (Start), the middle (Interim-Update), or the end (Stop), and whether the accounting function is on or off (Accounting-On or Accounting-Off). | 40 | integer | <ul style="list-style-type: none"> • Start (1) • Interim-Update • Stop (2) • On • Off |
| Acct-Session-Id | Either the Call-ID field value of the SIP INVITE message, the callIdentifier of the H.323 message, or RADIUS client information. | 44 | string | <ul style="list-style-type: none"> • Start • Interim-Update • Stop • On • Off |
| Acct-Session-Time | How much time in seconds (or milliseconds if so configured) the user has received service. | 46 | integer | <ul style="list-style-type: none"> • Interim-Update • Stop • Off |
| Acct-Terminate-Cause | How or why the session ended. | 49 | integer | <ul style="list-style-type: none"> • Stop • Off |

RADIUS Accounting Termination Causes

The table below describes the possible session termination causes for the Acct-Terminate-Cause RADIUS attribute.

| RADIUS Termination Cause | Related Integer Value (per RFC 2059) | Termination Event | Message |
|--------------------------|--------------------------------------|---|---------|
| User Request | 1 | A SIP BYE message. | Stop |
| User Error | 17 | Input from user is erroneous; for example, SIP signaling failed to establish the session. Used in combination with the Cisco Systems Disconnect Cause. (This termination cause is not used for H.323.) | Stop |
| Lost Service | 3 | Service cannot be sustained for reasons such as a lost connection. | Stop |
| idle-timeout | 4 | Idle timer expired. | Stop |
| session-timeout | 5 | Maximum session length timer expired. | Stop |
| Admin Reset | 6 | SBC hard reset occurred: A hard reset occurs when you use the front panel's orange Reset button; it reboots the SBC. | Off |
| Admin Reboot | 7 | SBC gracefully rebooted. | Off |
| NAS Request | 10 | RADIUS server is disabled; session terminated for non-error reason. | Off |

Cisco Systems RADIUS Decodes

The following table is a dictionary of the Cisco Systems (vendor identification number is 9) accounting VSAs. These attribute names are vendor-specific and subject to change without notice.

You can use the information in this table to translate the Cisco Systems VSAs that sometimes appear in SBC RADIUS messages into a more human-readable form.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|------------------|--|-----------------|----------------------|-------------------------------------|
| Setup Time | The SETUP message is used to request a connection (and therefore corresponds with the SIP INVITE). Start record—Time that the first SIP INVITE or H.323 SETUP message was received. Interim record—Time that the applicable SIP REINVITE message was received. Stop record—Time that the last SIP REINVITE message was received. | 25 | string | Start Interim Stop |
| Connect Time | Time that a SIP or H.323 session was accepted. This is the time a 200 OK SIP response to the SIP INVITE message was received or the time that a call ANSWERED/CONNECTED response to the H.323 SETUP message was received. | 28 | string | Start Interim- Update Stop |
| Disconnect Time | Time that a SIP BYE or H.323 Release Complete message was received or the session terminated. This is the time a SIP INVITE or H.323 SETUP transaction terminates for any reason. | 29 | string | Stop |
| Disconnect Cause | SIP Reasons for Disconnection (normal, redirection, client error, network error, global error, time-out, or user abandon) or the H.323 Release Complete Reason code (bad format address, unavailable, destination rejection, adaptive busy, etc.). For more information, refer to this guide's Mappings and Disconnect Cause Values section. | 30 | string | Stop |

Oracle RADIUS VSAs

Oracle's vendor identification number is 9148. This number refers to the 4-octet VSA Vendor-ID field. The high-order octet is 0 and the low-order 3 octets are the SMI Network Management Private Enterprise Code of the Vendor in network byte order, defined in the Assigned Numbers RFC (<http://www.fags.org/rfcs/rfc1700.html>; Reynolds, J. and J. Postel, Assigned Numbers, STD 2, RFC 1700, October 1994).

The table in this section is a dictionary of Oracle's accounting VSAs. You can use this information to translate the Oracle VSAs in SBC RADIUS messages into human-readable form. Oracle maintains VSA dictionary definition files for the most popular RADIUS distributions; ask your Oracle account representative for details.

Grouped according to attribute function, this table contains the following sections:

- **General Flow Attributes**—Overall traits of the media flow, these attributes appear in all CDRs regardless of the session's protocol; these attribute fields are only populated if there are media flows

- Inbound Flow Attributes—Detailed traits of the inbound media flow (including realm, remote IP address and port, and local IP address and port); these attribute fields are only populated if there are media flows
- Outbound Flow Attributes—Detailed traits of the outbound media flow (including realm, remote IP address and port, and local IP address and port); these attribute field are only populated if there are media flows
- Session Attributes—Information about the protocol type, ingress and egress realms used, and an identifier that links the H.323 and SIP legs of a call requiring IWF. In addition, SIP reporting includes specific information for Short Message Service (SMS) traffic, defined within the SBC as message events reported using CDR STOP records. SIP reporting also includes detail on VoLTE sessions to support management within IMS constructs.
- QoS Attributes—RADIUS call records are instantiated by individual signaling applications on the SBC. The SBC writes the following additional parameters to the call record for QoS (Quality of Service):
 - MSRP Total Packets
 - MSRP Total Octets
 - RTP Lost packets
 - RTP Jitter
 - RTP Maximum Jitter
 - RTCP Lost packets
 - RTCP Jitter
 - RTCP Latency
 - RTCP Maximum Latency
 - RTP Total Packets
 - RTP Total Octets

Only RADIUS Stop records contain QoS information. For non-QoS calls, the attributes appear in the record, but their values are always be zero (0). When you review the list of QoS VSAs, please note that “calling” in the attribute name means the information is sent by the calling party and called in the attribute name means the information is sent by the called party.

Examples of how this information appears in CDRs appears in Appendix B of this guide. Please note that the contents of Interim-Update messages do not depend on what events cause a Start message to be generated.

R-Factor and MOS

The SBC reports R-Factor and MOS data for the calling and called segments at the end of a session. This information appears in RADIUS CDRs, and in the Oracle VSA dictionary:

- Acme-Calling-R-Factor (151)
- Acme-Calling-MOS (152)
- Acme-Called-R-Factor (153)
- Acme-Called-MOS (154)

Note

These values are reported as * 100 in order to appear as integers.

Media Flow Attributes

The SBC records media flow attributes in RADIUS CDRs, and there can be multiple flows per session. In order to distinguish between the two flows that appear for a basic session (forward and reverse), the SBC supports unique media flow attribute names.

The term flow-set represents a pair of media flows, where one is the forward flow and one is the reverse. The flow attributes described in the table below have the designation FS1 or FS2, which identifies it as either the first or the second flow-set. In addition, all non-QoS attributes have a direction indicator: F for forward, and R for reverse.

MSRP Attributes

An additional group of MSRP media flow attributes are captured in the Acme-Extended-Attributes VSA. See [Oracle RADIUS Acme-Extended-Attributes VSAs](#) for information on configuring them.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|--------------------------|--|-----------------|----------------------|---------------------------------|
| Acme-CDR-Sequence-Number | Sequence number (that increases by 1) the SBC generates; recorded in each CDR. | 59 | integer | Start Interim-Update Stop |
| Acme-Intermediate-Time | Time interval at which periodic interim records are generated during a call. | 63 | string | Interim-Update |
| Acme-Local-Time-Zone | Local GMT/UTC time zone that is provisioned on the SBC. | 57 | string | Start Interim-Update Stop |
| Acme-Firmware-Version | Current software version running on the SBC. | 56 | string | Start Interim-Update Stop |

This table lists and describes general flow attributes.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-------------------|---|-----------------|----------------------|---------------------------------|
| Acme-FlowID_FS1_F | Unique identifier for every media flow processed by the SBC, flow-set 1 forward direction. This VSA always prefaces other flow information. | 1 | string | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------|---|-----------------|----------------------|---------------------------------|
| Acme-FlowID_FS1_R | Unique identifier for every media flow processed by the SBC, flow-set 1 reverse direction. This VSA always prefaces other flow information. | 78 | string | Start Interim-Update Stop |
| Acme-FlowID_FS2_F | Unique identifier for every media flow processed by the SBC, flow-set 2 forward direction. This VSA always prefaces other flow information. | 90 | string | Start Interim-Update Stop |
| Acme-FlowID_FS2_R | Unique identifier for every media flow processed by the SBC, flow-set 2 reverse direction. This VSA always prefaces other flow information. | 112 | string | Start Interim-Update Stop |
| Acme-FlowType_FS1_F | Codec that describes the flow, flow-set 1 forward direction: PCMU, PCMA, G722, G726, G723, G728, G729, H261, H263, T38. | 2 | string | Start Interim-Update Stop |
| Acme-FlowType_FS1_R | Codec that describes the flow, flow-set 1 reverse direction: PCMU, PCMA, G726, G723, G728, G729, H261, H263, T38. | 79 | string | Start Interim-Update Stop |
| Acme-FlowType_FS2_F | Codec that describes the flow, flow-set 2 forward direction: PCMU, PCMA, G726, G723, G728, G729, H261, H263, T38. | 91 | string | Start Interim-Update Stop |
| Acme-FlowType_FS2_R | Codec that describes the flow, flow-set 2 reverse direction: PCMU, PCMA, G726, G723, G728, G729, H261, H263, T38. | 113 | string | Start Interim-Update Stop |

This table describes inbound flow attributes.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------|--|-----------------|----------------------|---------------------------------|
| Acme-Flow-In-Realm_FS1_F | Inbound realm identifier for flow-set 1, forward direction. | 10 | string | Start Interim-Update Stop |
| Acme-Flow-In-Realm_FS1_R | Inbound realm identifier for flow-set 1, reverse direction. | 80 | string | Start Interim-Update Stop |
| Acme-Flow-In-Realm_FS2_F | Inbound realm identifier for flow-set 2, forward direction. | 92 | string | Start Interim-Update Stop |
| Acme-Flow-In-Realm_FS2_R | Inbound realm identifier for flow-set 2, reverse direction. | 114 | string | Start Interim-Update Stop |
| Acme-Flow-In-Src-Addr_FS1_F | Inbound source address (remote) information for flow-set 1, forward direction. | 11 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Src-Addr_FS1_R | Inbound source address (remote) information for flow-set 1, reverse direction. | 81 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Src-Addr_FS2_F | Inbound source address (remote) information for flow-set 2, forward direction. | 93 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Src-Addr_FS2_R | Inbound source address (remote) information for flow-set 2, reverse direction. | 115 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Src-Port_FS1_F | Inbound source (remote) port information for flow-set 1, forward direction. | 12 | integer | Start Interim-Update Stop |
| Acme-Flow-In-Src-Port_FS1_R | Inbound source (remote) port information for flow-set 1, reverse direction. | 82 | integer | Start Interim-Update Stop |
| Acme-Flow-In-Src-Port_FS2_F | Inbound source (remote) port information for flow-set 2, forward direction. | 94 | integer | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Flow-In-Src-Port_FS2_R | Inbound source (remote) port information for flow-set 2, reverse direction. | 116 | integer | Start Interim-Update Stop |
| Acme-Flow-In-Dst-Addr_FS1_F | Inbound destination (local) address information (the IPv4 address field value of the steering pool configuration) for flow-set 1, forward direction. | 13 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Dst-Addr_FS1_R | Inbound destination (local) address information (the IPv4 address field value of the steering pool configuration) for flow-set 1, reverse direction. | 83 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Dst-Addr_FS2_F | Inbound destination (local) address information (the IPv4 address field value of the steering pool configuration) for flow-set 2, forward direction. | 95 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Dst-Addr_FS2_R | Inbound destination (local) address information (the IPv4 address field value of the steering pool configuration) for flow-set 2, reverse direction. | 117 | IP address | Start Interim-Update Stop |
| Acme-Flow-In-Dst-Port_FS1_F | Inbound destination (local) port information (a port in the range between the start port and end port field values of the steering pool configuration) for flow-set 1, forward direction. | 14 | integer | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Flow-In-Dst-Port_FS1_R | Inbound destination (local) port information (a port in the range between the start port and end port field values of the steering pool configuration) for flow-set 1, reverse direction. | 84 | integer | Start Interim-Update Stop |
| Acme-Flow-In-Dst-Port_FS2_F | Inbound destination (local) port information (a port in the range between the start port and end port field values of the steering pool configuration) for flow-set 2, forward direction. | 96 | integer | Start Interim-Update Stop |
| Acme-Flow-In-Dst-Port_FS2_R | Inbound destination (local) port information (a port in the range between the start port and end port field values of the steering pool configuration) for flow-set 2, reverse direction. | 118 | integer | Start Interim-Update Stop |

This table lists and describes outbound flow attributes.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------|--|-----------------|----------------------|---------------------------------|
| Acme-Flow-Out-Realm_FS1_F | Outbound realm identifier for flow-set 1, forward direction. | 20 | string | Start Interim-Update Stop |
| Acme-Flow-Out-Realm_FS1_R | Outbound realm identifier for flow-set 1, reverse direction. | 85 | string | Start Interim-Update Stop |
| Acme-Flow-Out-Realm_FS2_F | Outbound realm identifier for flow-set 2, forward direction. | 97 | string | Start Interim-Update Stop |
| Acme-Flow-Out-Realm_FS2_R | Outbound realm identifier for flow-set 2, reverse direction. | 119 | string | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|----------------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Flow-Out- Src-Addr_FS1_F | Outbound source (local) address information (the IPv4 address field value of the steering port configuration) for flow-set 1, forward direction. | 21 | IP address | Start Interim-Update Stop |
| Acme-Flow-Out- Src-Addr_FS1_R | Outbound source (local) address information (the IPv4 address field value of the steering port configuration) for flow-set 1, reverse direction. | 86 | IP address | Start Interim-Update Stop |
| Acme-Flow-Out- Src-Addr_FS2_F | Outbound source (local) address information (the IPv4 address field value of the steering port configuration) for flow-set 2, forward direction. | 98 | IP address | Start Interim-Update Stop |
| Acme-Flow-Out- Src-Addr_FS2_R | Outbound source (local) address information (the IPv4 address field value of the steering port configuration) for flow-set 2, reverse direction. | 120 | IP address | Start Interim-Update Stop |
| Acme-Flow-Out- Src-Port_FS1_F | Outbound source (local) port information for flow-set 1, forward direction (a port in the range between the start port and end port field values of the steering port configuration). | 22 | integer | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|----------------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Flow-Out- Src-Port_FS1_R | Outbound source (local) port information for flow-set 1, reverse direction (a port in the range between the start port and end port field values of the steering port configuration). | 87 | integer | Start Interim-Update Stop |
| Acme-Flow-Out- Src-Port_FS2_F | Outbound source (local) port information for flow-set 2, forward direction (a port in the range between the start port and end port field values of the steering port configuration). | 99 | integer | Start Interim-Update Stop |
| Acme-Flow-Out- Src-Port_FS2_R | Outbound source (local) port information for flow-set 2, reverse direction (a port in the range between the start port and end port field values of the steering port configuration). | 121 | integer | Start Interim-Update Stop |
| Acme-Flow-Out- Dst-Addr_FS1_F | Outbound destination (remote) address information for flow-set 1, forward direction. | 23 | IP address | Start Interim-Update Stop |
| Acme-Flow-Out- Dst-Addr_FS1_R | Outbound destination (remote) address information for flow-set 1, reverse direction. | 88 | IP address | Start Interim-Update Stop |
| Acme-Flow-Out- Dst-Addr_FS2_F | Outbound destination (remote) address information for flow-set 2, forward direction. | 100 | IP address | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|------------------------------|--|-----------------|----------------------|---------------------------------|
| Acme-Flow-Out-Dst-Addr_FS2_R | Outbound destination (remote) address information for flow-set 2, reverse direction. | 122 | IP address | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-Port_FS1_F | Outbound destination (remote) port information for flow-set 1, forward direction. | 24 | integer | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-Port_FS1_R | Outbound destination (remote) port information for flow-set 1, reverse direction. | 89 | integer | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-Port_FS2_F | Outbound destination (remote) port information for flow-set 2, forward direction. | 101 | integer | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-Port_FS2_R | Outbound destination (remote) port information for flow-set 2, reverse direction. | 123 | integer | Start Interim-Update Stop |

This table lists and describes session attributes.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Session- Generic-Id | Common ID shared by H.323 and SIP call legs of a session. This attribute is a combination of a time stamp (measured in seconds) and a monotonically increasing 16-bit integer, followed by an at-sign (@) and the MAC address of the rear interface (wancom). This attribute is only used to correlate the H.323 and SIP legs of an interworking call/session. This VSA is not configurable; all CDRs contain this attribute. | 40 | string | Start Interim-Update Stop |
| Acme-Session- Ingress-CallId | Call ID generated by the originating device. | 3 | string | Start Interim-Update Stop |
| Acme-Session- Egress-CallId | Call ID generated by the SBC to represent a two-way transaction. | 4 | string | Start Interim-Update Stop |
| Acme-Session- Ingress-Realm | Explicitly identifies the ingress realm, and contains the name of the ingress realm for the session. All CDRs contain this attribute. This VSA is not configurable; all CDRs contain this attribute. | 41 | string | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|------------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Session-Egress-Realm | Explicitly identifies the egress realm, and contains the name of the egress realm for the session. All CDRs contain this attribute. This VSA is not configurable. All CDRs contain this attribute, but it is only populated if an egress realm is found; a call without a route does not have an egress realm. | 42 | string | Start Interim-Update Stop |
| Acme-Session-Protocol-Type | Signaling protocol used for a particular leg of a session (in the case of IWF, there may be two legs). This attribute contains the signaling protocol type; for example, SIP or H323. This VSA is not configurable; all CDRs contain this attribute. | 43 | string | Start Interim-Update Stop |
| Acme-Session-Charging-Vector | Appears when the SBC inserts, passes, or deletes the P-Charging-Vector header (SIP). This attribute is only populated for SIP CDRs, and is not populated if the SBC does not have P-Charging-Vector information. | 54 | string | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|--|---|-----------------|----------------------|---------------------------------|
| Acme-Session-Charging-Function_Address | Appears when the SBC inserts, passes, or deletes the P-Charging-Function-Address. This attribute is only populated for SIP CDRs, and is not populated if the SBC does not have P-Charging-Function-Address information. | 55 | string | Start Interim-Update Stop |
| Acme-Session-Disposition | Status of the call attempt as it progresses from being initiated (using a SIP INVITE or H.323 Setup message) to being either answered or failing to be answered. | 60 | integer | Start Interim-Update Stop |
| Acme-Post-Dial-Delay | Amount of time between session initiation and an alerting event. | 58 | integer | Start Interim-Update Stop |
| Acme-P-Asserted-ID | P-Asserted ID as described in RFC 3325. | 69 | string | Start Interim-Update Stop |
| Acme-SIP-Diversion | SIP Diversion header; communicates to the called party from whom and why a call diverted. | 70 | string | Start Interim-Update Stop |
| Acme-Primary-Routing-Number | Primary routing number and phone context (or ingress SIP Request-URI). | 64 | string | Start Interim-Update Stop |
| Acme-Egress-Final-Routing-Number | Final routing number and phone context (or egress SIP Request-URI). | 134 | integer | Stop |
| Acme-Disconnect-Initiator | Initiator of a call disconnect. | 61 | integer | Stop |
| Acme-Disconnect-Cause | Q.850 cause code value. | 62 | integer | Stop |
| Acme-SIP-Status | SIP status code for RFC 3326 support. | 71 | integer | Stop |
| Acme-Originating-Trunk-Group | Originating trunk group. | 65 | string | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Originating-Trunk-Context | Originating trunk group context. | 67 | string | Start Interim-Update Stop |
| Acme-Terminating-Trunk-Group | Terminating trunk group. | 66 | string | Start Interim-Update Stop |
| Acme-Terminating-Trunk-Context | Terminating trunk group context. | 68 | string | Start Interim-Update Stop |
| Acme-Ingress-Local-Addr | Signaling IP address and port of the ingress SBC signaling interface. | 74 | string | Start Interim-Update Stop |
| Acme-Ingress-Remote-Addr | Signaling IP address and port of the ingress remote signaling element. | 75 | string | Start Interim-Update Stop |
| Acme-Egress-Local-Addr | Signaling IP address and port of the egress SBC signaling interface. | 76 | string | Start Interim-Update Stop |
| Acme-Egress-Remote-Addr | Signaling IP address and port of the destination signaling element. | 77 | string | Start Interim-Update Stop |
| Acme-Session-Ingress-RPH | RPH value received in the incoming call (e.g., ets.1). Only populated for NSEP calls. | 135 | string | Start Interim-Update Stop |
| Acme-Session-Egress-RPH | RPH value sent in the outgoing call (e.g., ets.3). Only populated for NSEP calls. | 136 | string | Start Interim-Update Stop |
| Acme-Ingress-Network-Interface-Id | To differentiate overlapping IP address spaces (with the Acme-Ingress-Vlan-Tag-Value), gives the ID of the ingress network interface. | 137 | string | Start Interim-Update Stop |
| Acme-Ingress-Vlan-Tag-Value | To differentiate overlapping IP address spaces (with the Acme-Ingress-Network-Interface-Id), gives the VLAN tag. | 138 | integer | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|----------------------------------|--|-----------------|----------------------|---------------------------------|
| Acme-Egress-Network-Interface-Id | To differentiate overlapping IP address spaces (with the Acme-Egress-Vlan-Tag-Value), gives the ID of the ingress network interface. | 139 | string | Start Interim-Update Stop |
| Acme-Egress-Vlan-Tag-Value | To differentiate overlapping IP address spaces (with the Acme-Egress-Network-Interface-Id), gives the VLAN tag. | 140 | integer | Start Interim-Update Stop |
| Acme-Refer-Call-Transfer-Id | For SIP REFER call method transfer, communicates a call has been transferred from the referer to the referree | 141 | string | Stop |
| Acme-Extended-Attributes-1 | Indicates the name, tone family, and call side of any VBD tone detected on the call. | 182 | string | Interim-Update Stop |

This table lists and describes QoS attributes.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|------------------------------------|---|-----------------|----------------------|----------|
| Acme-Calling-RTCP-Packets-Lost_FS1 | Total lost packets reported via Real-time Transport Protocol Control Protocol (RTCP), flow-set 1. Populated only if QoS is enabled. | 32 | integer | Stop |
| Acme-Calling-RTCP-Packets-Lost_FS2 | Total lost packets reported via Real-time Transport Protocol Control Protocol (RTCP), flow-set 2. Populated only if QoS is enabled. | 104 | integer | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------------|--|-----------------|----------------------|----------|
| Acme-Calling-RTCP-Avg-Jitter_FS1 | Average jitter reported via RTCP measured in milliseconds, flow-set 1. Populated only if QoS is enabled. | 33 | integer | Stop |
| Acme-Calling-RTCP-Avg-Jitter_FS2 | Average jitter reported via RTCP measured in milliseconds, flow-set 2. Populated only if QoS is enabled. | 105 | integer | Stop |
| Acme-Calling-RTCP-Avg-Latency_FS1 | Average latency reported by comparing the timestamps in RTCP packets for each direction of a call, flow-set 1. Populated only if QoS is enabled. | 34 | integer | Stop |
| Acme-Calling-RTCP-Avg-Latency_FS2 | Average latency reported by comparing the timestamps in RTCP packets for each direction of a call, flow-set 2. Populated only if QoS is enabled. | 106 | integer | Stop |
| Acme-Calling-RTCP-MaxJitter_FS1 | Maximum amount of jitter value reported via RTCP measured in milliseconds, flow-set 1. Populated only if QoS is enabled. | 35 | integer | Stop |
| Acme-Calling-RTCP-MaxJitter_FS2 | Maximum amount of jitter value reported via RTCP measured in milliseconds, flow-set 3. Populated only if QoS is enabled. | 107 | integer | Stop |
| Acme-Calling-RTCP-MaxLatency_FS1 | Maximum latency value measured in milliseconds as observed through RTCP, flow-set 1. Populated only if QoS is enabled. | 36 | integer | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------------|--|-----------------|----------------------|----------|
| Acme-Calling-RTCP-MaxLatency_FS2 | Maximum latency value measured in milliseconds as observed through RTCP, flow-set 2. Populated only if QoS is enabled. | 108 | integer | Stop |
| Acme-Calling-Octets_FS1 | Bytes of RTP traffic for this call, flow-set 1. Populated only if QoS is enabled. | 28 | integer | Stop |
| Acme-Calling-Octets_FS2 | Bytes of RTP traffic for this call, flow-set 2. Populated only if QoS is enabled. | 102 | integer | Stop |
| Acme-Calling-Packets_FS1 | Number of RTP Packets, received by the SBC, from the calling party, for flow-set 1. Populated only if QoS is enabled. | 29 | integer | Stop |
| Acme-Calling-Packets_FS2 | Number of RTP Packets, received by the SBC, from the calling party, for flow-set 2. Populated only if QoS is enabled. | 103 | integer | Stop |
| Acme-Calling-RTP-Packets-Lost_FS1 | Total RTP packets lost in flow-set 1. Populated only if QoS is enabled. | 37 | integer | Stop |
| Acme-Calling-RTP-Packets-Lost_FS2 | Total RTP packets lost in flow-set 2. Populated only if QoS is enabled. | 109 | integer | Stop |
| Acme-Calling-RTP-Avg-Jitter_FS1 | Total jitter measured on RTP packets in milliseconds, flow-set 1. Populated only if QoS is enabled. | 38 | integer | Stop |
| Acme-Calling-RTP-Avg-Jitter_FS2 | Total jitter measured on RTP packets in milliseconds, flow-set 2. Populated only if QoS is enabled. | 110 | integer | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---|---|-----------------|----------------------|----------|
| Acme-Calling-RTP- MaxJitter_FS1 | Maximum jitter measured on RTP packets in milliseconds, flow-set 1. Populated only if QoS is enabled. | 39 | integer | Stop |
| Acme-Calling-RTP- Avg- MaxJitter_FS2 | Maximum jitter measured on RTP packets in milliseconds, flow-set 2. Populated only if QoS is enabled. | 111 | integer | Stop |
| Acme-Called- Octets_FS1 | Number of Octets (8 bits) of RTP traffic, received by the SBC, from the called party, for flow-set 1. Populated only if QoS is enabled. | 44 | integer | Stop |
| Acme-Called- Octets_FS2 | Number of Octets (8 bits) of RTP traffic, received by the SBC, from the called party, for flow-set 2 . Populated only if QoS is enabled. | 124 | integer | Stop |
| Acme-Called- Packets_FS1 | Number of RTP Packets, received by the SBC, from the called party, for flow-set 1. Populated only if QoS is enabled. | 45 | integer | Stop |
| Acme-Called- Packets_FS2 | Number of RTP Packets, received by the SBC, from the called party, for flow-set 2 . Populated only if QoS is enabled. | 125 | integer | Stop |
| Acme-Called- RTCP-Packets- Lost_FS1 | Total lost packets reported via Real-time Transport Protocol Control Protocol (RTCP), flow-set 1. Populated only if QoS is enabled. | 46 | integer | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------------|---|-----------------|----------------------|----------|
| Acme-Called-RTCP-Packets-Lost_FS2 | Total lost packets reported via Real-time Transport Protocol Control Protocol (RTCP), flow-set 2. Populated only if QoS is enabled. | 126 | integer | Stop |
| Acme-Called-RTCP-Avg-Jitter_FS1 | Average jitter reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 1. Populated only if QoS is enabled. | 47 | integer | Stop |
| Acme-Called-RTCP-Avg-Jitter_FS2 | Average jitter reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 2. Populated only if QoS is enabled. | 127 | integer | Stop |
| Acme-Called-Avg-Latency_FS1 | Average latency reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 1. Populated only if QoS is enabled. | 48 | integer | Stop |
| Acme-Called-Avg-Latency_FS2 | Average latency reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 2. Populated only if QoS is enabled. | 128 | integer | Stop |
| Acme-Called-RTCP-MaxJitter_FS1 | Maximum amount of jitter reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 1. Populated only if QoS is enabled. | 49 | integer | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|----------------------------------|--|-----------------|----------------------|----------|
| Acme-Called-RTCP-MaxJitter_FS2 | Maximum amount of jitter reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 2. Populated only if QoS is enabled. | 129 | integer | Stop |
| Acme-Called-RTCP-MaxLatency_FS1 | Maximum amount of latency reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 1. Populated only if QoS is enabled. | 50 | integer | Stop |
| Acme-Called-RTCP-MaxLatency_FS2 | Maximum amount of latency reported via RTCP measured in milliseconds for the ingress side of the call, flow-set 2. Populated only if QoS is enabled. | 130 | integer | Stop |
| Acme-Called-RTP-Packets-Lost_FS1 | Total lost RTP packets for the ingress side of the call, flow-set 1. Populated only if QoS is enabled. | 51 | integer | Stop |
| Acme-Called-RTP-Packets-Lost_FS2 | Total lost RTP packets for the ingress side of the call, flow-set 2. Populated only if QoS is enabled. | 131 | integer | Stop |
| Acme-Called-RTP-Avg-Jitter_FS1 | Average jitter reported via RTP measured in milliseconds for the ingress side of the realm, flow-set 1. Populated only if QoS is enabled. | 52 | integer | Stop |
| Acme-Called-RTP-Avg-Jitter_FS2 | Average jitter reported via RTP measured in milliseconds for the ingress side of the realm, flow-set 2. Populated only if QoS is enabled. | 132 | integer | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-------------------------------|---|-----------------|----------------------|----------|
| Acme-Called-RTP-MaxJitter_FS1 | Maximum amount of jitter reported via RTP measured in milliseconds for the ingress side of the call, flow-set1. Populated only if QoS is enabled. | 53 | integer | Stop |
| Acme-Called-RTP-MaxJitter_FS2 | Maximum amount of jitter reported via RTP measured in milliseconds for the ingress side of the call, flow-set 2. Populated only if QoS is enabled. | 133 | integer | Stop |
| Acme-Calling-R-Factor | QoS R-Factor calculation for the calling side of a session. Populated only if QoS is enabled. This value is reported as * 100 in order to appear as an integer. | 151 | integer | Stop |
| Acme-Calling-MOS | QoS MOS calculation for the calling side of a session. Populated only if QoS is enabled. This value is reported as * 100 in order to appear as an integer. | 152 | integer | Stop |
| Acme-Called-R-Factor | QoS R-Factor calculation for the called side of a session. Populated only if QoS is enabled. This value is reported as * 100 in order to appear as an integer. | 153 | integer | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------------------|--|-----------------|----------------------|---|
| Acme-Called-MOS New in Release | QoS MOS calculation for the called side of a session. Populated only if QoS is enabled. This value is reported as * 100 in order to appear as an integer. | 154 | integer | Stop |
| Acme-Session-Forked-Call-Id | The VSA is a string value, and appears as the header-value without the header parameters from the P-Multiring-Correlator header for a session identified as part of a forked call. | 171 | string | Stop |
| Acme-Flow-Calling-Media-Stop-Time_FS1 | calling side's media stop time - stream 1 | 231 | string | Start Interim-Update Interim-Update (error) Stop |
| Acme-Flow-Called-Media-Stop-Time_FS1 | called side's media stop time - stream 1 | 232 | string | Start Interim-Update Interim-Update (error) Stop |
| Acme-Flow-Calling-Media-Stop-Time_FS2 | calling side's media stop time - stream 2 | 233 | string | Start Interim-Update Interim-Update (error) Stop |
| Acme-Flow-Called-Media-Stop-Time_FS2 | called side's media stop time - stream 2 | 234 | string | Start Interim-Update Interim-Update (error) Stop |

IPv6 Support

The following table lists the media flow attributes for IPv6 flows.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|-----------------------------------|--|-----------------|----------------------|---------------------------------|
| Acme-Flow-In-Src-IPv6_Addr_FS1_F | Inbound source IPv6 address (remote) information for flow-set 1, forward direction. | 155 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-In-Dst-IPv6_Addr_FS1_F | Inbound destination (local) address information (the IPv6 address field value of the steering pool configuration) for flow-set 1, forward direction. | 156 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Src-IPv6_Addr_FS1_F | Outbound source (local) address information (the IPv6 address field value of the steering port configuration) for flow-set 1, forward direction. | 157 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-IPv6_Addr_FS1_F | Outbound destination (remote) IPv6 address information for flow-set 1, forward direction. | 158 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-In-Src-IPv6_Addr_FS1_R | Inbound source IPv6 address (remote) information for flow-set 1, reverse direction. | 159 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-In-Dst-IPv6_Addr_FS1_R | Inbound destination (local) address information (the IPv6 address field value of the steering pool configuration) for flow-set 1, reverse direction. | 160 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Src-IPv6_Addr_FS1_R | Outbound source (local) address information (the IPv6 address field value of the steering port configuration) for flow-set 1, reverse direction. | 161 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-IPv6_Addr_FS1_R | Outbound destination (remote) IPv6 address information for flow-set 1, reverse direction. | 162 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-In-Src-IPv6_Addr_FS2_F | Inbound source address (remote) IPv6 information for flow-set 2, forward direction. | 163 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-In-Dst-IPv6_Addr_FS2_F | Inbound destination (local) address information (the IPv6 address field value of the steering pool configuration) for flow-set 2, forward direction. | 164 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Src-IPv6_Addr_FS2_F | Outbound source (local) address information (the IPv6 address field value of the steering port configuration) for flow-set 2, forward direction. | 165 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-IPv6_Addr_FS2_F | Outbound destination (remote) IPv6 address information for flow-set 2, forward direction. | 166 | ipv6addr | Start Interim-Update Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------------------|--|-----------------|----------------------|---|
| Acme-Flow-In-Src-IPv6_Addr_FS2_R | Inbound source address (remote) IPv6 address information for flow-set 2, reverse direction. | 167 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-In-Dst-IPv6_Addr_FS2_R | Inbound destination (local) address information (the IPv6 address field value of the steering pool configuration) for flow-set 2, reverse direction. | 168 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Src-IPv6_Addr_FS2_R | Outbound source (local) address information (the IPv6 address field value of the steering port configuration) for flow-set 2, reverse direction. | 169 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Out-Dst-IPv6_Addr_FS2_R | Outbound destination (remote) IPv6 address information for flow-set 2, reverse direction. | 170 | ipv6addr | Start Interim-Update Stop |
| Acme-Flow-Calling-Media-Stop-Time_FS1 | calling side's media stop time - stream 1 | 231 | string | Start Interim-Update Interim-Update (error) Stop |
| Acme-Flow-Called-Media-Stop-Time_FS1 | called side's media stop time - stream 1 | 232 | string | Start Interim-Update Interim-Update (error) Stop |
| Acme-Flow-Calling-Media-Stop-Time_FS2 | calling side's media stop time - stream 2 | 233 | string | Start Interim-Update Interim-Update (error) Stop |
| Acme-Flow-Called-Media-Stop-Time_FS2 | called side's media stop time - stream 2 | 234 | string | Start Interim-Update Interim-Update (error) Stop |

Oracle VSA Values

The table below defines the possible values for several Oracle VSAs.

| Oracle VSA Name | Attribute Value | Possible Values |
|--------------------------|-----------------|--|
| Acme-PostDial-Delay | 58 | Unit value in milliseconds |
| Acme-Session-Disposition | 60 | 0=unknown 1=call_attempt 2=ringing 3=answered |

| Oracle VSA Name | Attribute Value | Possible Values |
|---------------------------|-----------------|---|
| Acme-Disconnect-Initiator | 61 | 0=UNKNOWN_DISCONNECT_INITIATOR 1=CALLING_PARTY_DISCONNECT 2=CALLED_PARTY_DISCONNECT 3=INTERNAL_DISCONNECT |
| Acme-Disconnect-Cause | 62 | 34=No circuit/channel available 47=Resource unavailable 3=No route destination 31=Normal, unspecified 88=Incompatible destination 111=Interworking, unspecified 38=Network out of order 42=Switching equip congestion 28=Invalid number format 41=Temporary failure 17=User busy 16=Normal call clearing 20=Subscriber absent 31=Normal call clearing 18=Request error timeout response 55=Forbidden error response 0=No reason cause presented |
| Acme-SIP-Diversion | 70 | SIP Diversion header based on this RFC draft: draft-levy-sip-diversion-05.txt |

| Oracle VSA Name | Attribute Value | Possible Values |
|-----------------|-----------------|---|
| Acme-SIP-Status | 71 | <p>This is a complete list of support status codes; only a subset would be reported in a Stop record:</p> <p>RESP_STATUS_TRYING 100</p> <p>RESP_STATUS_RINGING 180</p> <p>RESP_STATUS_FORWARD 181</p> <p>RESP_STATUS_QUEUED 182</p> <p>RESP_STATUS_PROGRESS 183</p> <p>RESP_STATUS_OK 200</p> <p>RESP_STATUS_CREATED 201</p> <p>RESP_STATUS_ACCEPTED 202</p> <p>RESP_STATUS_PART 206</p> <p>RESP_STATUS_MAX_OK 299</p> <p>RESP_STATUS_MULTIPLE 300</p> <p>RESP_STATUS_MOVED 301</p> <p>RESP_STATUS_MOVED_TMP 302</p> <p>RESP_STATUS_USE_PROXY 305</p> <p>RESP_STATUS_ALTERNATE 380</p> <p>RESP_STATUS_BAD 400</p> <p>RESP_STATUS_UNAUTH 401</p> <p>RESP_STATUS_PAY_REQ 402</p> <p>RESP_STATUS_FORBIDDEN 403</p> <p>RESP_STATUS_NOT_FOUND 404</p> <p>RESP_STATUS_NOT_ALLOW 405</p> <p>RESP_STATUS_NOT_ACCEPT 406</p> <p>RESP_STATUS_AUTH_REQ 407</p> <p>RESP_STATUS_REQ_TMO 408</p> <p>RESP_STATUS_CONFLICT 409</p> <p>RESP_STATUS_GONE 410</p> <p>RESP_STATUS_LEN_REQ 411</p> <p>RESP_STATUS_TOO_BIG 413</p> <p>RESP_STATUS_URI_TOO_BIG 414</p> <p>RESP_STATUS_MEDIA 415</p> <p>RESP_STATUS_URI_SCHEME 416</p> <p>RESP_STATUS_BAD_EXT 420</p> <p>RESP_STATUS_EXT_REQ 421</p> <p>RESP_STATUS_TOO_SMALL 422</p> <p>RESP_STATUS_TOO_BRIEF 423</p> <p>RESP_STATUS_TMP_UNAVAIL 480</p> <p>RESP_STATUS_NO_EXIST 481</p> <p>RESP_STATUS_LOOP 482</p> <p>RESP_STATUS_TOOMNY_HOPS 483</p> <p>RESP_STATUS_ADDR_INCMPL 484</p> <p>RESP_STATUS_AMBIGUOUS 485</p> <p>RESP_STATUS_BUSY_HERE 486</p> <p>RESP_STATUS_CANCELLED 487</p> <p>RESP_STATUS_NOT_HERE 488</p> <p>RESP_STATUS_BAD_EVENT 489</p> |

| Oracle VSA Name | Attribute Value | Possible Values |
|----------------------------|-----------------|---|
| | | RESP_STATUS_PENDING 491 |
| | | RESP_STATUS_UNDECIPH 493 |
| | | RESP_STATUS_INT_ERR 500 |
| | | RESP_STATUS_NOT_IMPL 501 |
| | | RESP_STATUS_BAD_GTWY 502 |
| | | RESP_STATUS_SVC_UNAVAIL 503 |
| | | RESP_STATUS_GTWY_TMO 504 |
| | | RESP_STATUS_BAD_VER 505 |
| | | RESP_STATUS_MSG_TOO_BIG 513 |
| | | RESP_STATUS_PRE_FAIL 580 |
| | | RESP_STATUS_BUSY 600 |
| | | RESP_STATUS_DECLINE 603 |
| | | RESP_STATUS_DONT_EXIST 604 |
| | | RESP_STATUS_NOTACCEPT 606 |
| Acme-Extended-Attributes-1 | 182 | <p>The modem tones detected in a call. This is a complete list of possible values; only those detected on a call would be included in the Interim-Update or Stop message.</p> <p>Modem-Ans-ANS</p> <p>Modem-Ans-/ANS</p> <p>Modem-Ans-ANSam</p> <p>Modem-Ans-/ANSam</p> <p>Modem-Ans-Bell103</p> <p>Modem-Ans-V21High</p> <p>Modem-Ans-V23High</p> <p>Modem-Ans-V22</p> <p>Modem-Ans-V22+1800Hz</p> <p>Modem-Ans-V8bis</p> <p>Modem-Ans-Baudot</p> <p>Modem-Orig-Bell103</p> <p>Modem-Orig-V21Low</p> <p>Modem-Orig-V25Calling</p> <p>Modem-Orig-V8bis</p> <p>Modem-Orig-Baudot</p> |

Authentication VSAs

The table below defines Oracle VSAs used for RADIUS authentication.

| Oracle VSA Name | Attribute Value | Attribute Values |
|---------------------|---|------------------|
| Acme-User-Privilege | Describes at RADIUS login the privileges granted to the administrator (VSA only available with admin security license installed). Values can be: sftpForAudit (SFTP is allowed for audit logs) sftpForAll (SFTP is allowed for logging, and audit logs) | 253 |
| Acme-User-Class | Identifies the authorization class on the SBC; used for RADIUS authentication only and does not apply to accounting. Values can be user or admin | 254 |

RTP Traffic Reporting per Call

The SBC reports total RTP traffic counts, both received and transmitted for calls through standard accounting interfaces on stop record generation. This traffic is reported in Packets and Octets, sent and received, for flow-sets 1 and 2. The QoS feature set must be enabled to report on RTP traffic, otherwise the values will be reported as 0

These statics are captured for the following scenarios

- RTP pass-thru sessions
- transcoded/transrated/inband (audio) DTMF-interworked RTP sessions
- RTP sessions where one or both call legs is encrypted (SRTP)

RTP traffic reporting does not capture MSRP B2BUA and MSRP NAT traffic.

The quick way to decipher these 16 statistics are as follows:

- Calling/Called - call-leg the static reports on
- Octets/Packets - counter unit for traffic
- FS1/FS2 - flow set 1 or flow set 2
- blank/transmitted - traffic received (blank) or transmitted by the SBC

| Counter Definition | RADIUS/Local CDR (VSA #) output | AVP in Acme-Packet-Specific- Rf-QoS(37) |
|---|------------------------------------|--|
| Number of Octets (8 bits) of RTP traffic, received by the SBC, from the calling party, for flow-set 1 | Acme-Calling-Octets-FS1 (28) | RTP-Calling-Octets-FS1 (38) |
| Number of RTP Packets, received by the SBC, from the calling party, for flow-set 1 | Acme-Calling-Packets-FS1 (29) | RTP-Calling-Packets-FS1(40) |
| Number of Octets (8 bits) of RTP traffic, received by the SBC, from the called party, for flow-set 1 | Acme-Called-Octets-FS1 (44) | RTP-Called-Octets-FS1 (62) |
| Number of RTP Packets, received by the SBC, from the called party, for flow-set 1 | Acme-Called-Packets-FS1 (45) | RTP-Called-Packets-FS1 (64) |

| Counter Definition | RADIUS/Local CDR (VSA #) output | AVP in Acme-Packet-Specific- Rf-QoS(37) |
|--|--|--|
| Number of Octets (8 bits) of RTP traffic, received by the SBC, from the calling party, for flow-set 2 | Acme-Calling-Octets-FS2 (102) | RTP-Calling-Octets-FS2 (39) |
| Number of RTP Packets, received by the SBC, from the calling party, for flow-set 2 | Acme-Calling-Packets-FS2 (103) | RTP-Calling-Packets-FS2(41) |
| Number of Octets (8 bits) of RTP traffic, received by the SBC, from the called party, for flow-set 2 | Acme-Called-Octets-FS2 (124) | RTP-Called-Octets-FS2 (63) |
| Number of RTP Packets, received by the SBC, from the called party, for flow-set 2 | Acme-Called-Packets-FS2 (125) | RTP-Called-Packets-FS2 (65) |
| Number of Octets (8 bits) of RTP traffic, transmitted by the SBC, to the calling party, for flow-set 1 | Acme-Calling-RTP-Octet-Transmitted-FS1 (240) | RTP-Calling-Octets-Transmitted-FS1 (42) |
| Number of RTP Packets, transmitted by the SBC, to the calling UA, for flow-set 1 | Acme-Calling-RTP-Packets-Transmitted-FS1 (241) | RTP-Calling-Packets-Transmitted-FS1 (44) |
| Number of Octets (8 bits) of RTP traffic, transmitted by the SBC, to the called UA, for flow-set 1 | Acme-Called-RTP-Octet-Transmitted-FS1 (242) | RTP-Called-Octets-Transmitted-FS1 (66) |
| Number of RTP Packets, transmitted by the SBC, to the called UA, for flow-set 1 | Acme-Called-RTP-Packets-Transmitted-FS1 (243) | RTP-Called-Packets-Transmitted-FS1 (68) |
| Number of Octets (8 bits) of RTP traffic, transmitted by the SBC, to the calling party, for flow-set 2 | Acme-Calling-RTP-Octets-Transmitted-FS2 (244) | RTP-Calling-Octets-Transmitted-FS2 (43) |
| Number of RTP Packets, transmitted by the SBC, to the calling UA, for flow-set 2 | Acme-Calling-RTP-Packets-Transmitted-FS2 (245) | RTP-Calling-Packets-Transmitted-FS1 (45) |
| Number of Octets (8 bits) of RTP traffic, transmitted by the SBC, to the called UA, for flow-set 2 | Acme-Called-RTP-Octet-Transmitted-FS2 (246) | RTP-Called-Octets-Transmitted-FS2 (67) |
| Number of RTP Packets, transmitted by the SBC, to the called UA, for flow-set 2 | Acme-Called-RTP-Packets-Transmitted-FS2 (247) | RTP-Called-Packets-Transmitted-FS2 (69) |

VoLTE and SMS VSAs

The SBC reports session-specific information for VoLTE calls and for Short Message Service (SMS) messages. Much of this information overlaps both session types as they address similar variables within their environments.

This table lists and describes the VoLTE and SMS attributes and includes attribute name, attribute description, attribute value, attribute value type, and messages.

Note

See the [Oracle RADIUS Acme-Extended-Attributes VSAs](#) section for an explanation of the attribute extensions included in the tables below.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------------|---|-----------------|----------------------|---------------------------------|
| Acme-Access-Network-Information | Extracted from Access-Network-Information field from P-Access-Network-Info headers. For MO calls it should be the PANI headers of the outgoing INVITE (after the NPLI procedure). For MT calls it should be the PANI headers of the outgoing 18x response (after the NPLI procedure). | 248 | SMS and VoLTE | Start Interim-Update Stop |
| Acme-P-GW IP Address | Obtained from PCRF RAR/AAA in Access-Network-Charging-Address (501) AVP. | 249, ext 1 | VoLTE call | Start Interim-Update Stop |
| Acme-S-GW IP Address | Obtained from PCRF AAA/RAR in AN-GW-Address (1050) AVP | 249, ext 2 | VoLTE call | Start Interim-Update Stop |
| Acme-Originating-IOI | Extracted from the Originating-IOI field in the P-Charging-Vector header. For MT, MO (MESSAGE/INVITE) calls, the field is extracted from SIP reply(20X). | 249, ext 3 | SMS and VoLTE call | Start Interim-Update Stop |
| Acme-Terminating-IOI | Extracted from the Terminating-IOI field in the P-Charging-Vector header. For MT, MO (MESSAGE/INVITE) calls, the field is extracted from SIP reply(20X). | 249, ext 4 | SMS and VoLTE call | Start Interim-Update Stop |
| Acme-IMEI | Extracted from the registration cache or Initial request. (The Initial request takes priority.) | 249, ext 5 | SMS and VoLTE call | Start Interim-Update Stop |
| Acme-Node-Functionality | Configured with a single, global Node Functionality value. This is done in the SIP config's node functionality parameter. However, if the node functionality parameter is also configured in a realm config, the ingress realm's node functionality value supersedes the global value. | 249, ext 6 | SMS and VoLTE call | Start Interim-Update Stop |
| Acme-SMS Message Type | Extracted from initial SIP MESSAGE. | 249, ext 7 | SMS | Stop |
| Acme-SMS Calling party number | Extracted from initial SIP MESSAGE. For MO, from the P-Asserted-Identity header For MT, from the TP-Originating-Address | 249, ext 8 | SMS | Stop |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------------|--|-------------------|----------------------|---------------------------------|
| Acme-SMS Called party number | Extracted from initial SIP MESSAGE. For MO, from the TP-Destination-Address For MT, from the To header of the SIP MESSAGE | 249, extension 9 | SMS | Stop |
| Acme-Message Length | Extracted from SIP MESSAGE field TP-User-Data-Length | 249, extension 10 | SMS | Stop |
| Acme-History-Info | Extracted from History-Info sip headers, ingress interface and it taken from initial message. In case of multiple History-Info headers, concatenated into a single header values in CDR. | 250 | VoLTE call | Start Interim-Update Stop |
| Acme-Visited-Network-Identifier | Extracted from Visited-Network-Identifier field from P-Visited-Network-Id headers. For MO calls, the field is extracted from initial request, or from the ingress sip-interface if the PVNI is not received in the initial request. For MT calls, the field is extracted from the initial request. | 251 | SMS and VoLTE call | Start Interim-Update Stop |
| Acme-IMSI | Extracted from the registration cache or Initial request. (The Initial request takes priority.) | 252 | SMS and VoLTE call | Start Interim-Update Stop |

This information appears in RADIUS CDRs, CSV CDRs, and the Oracle VSA dictionary. The SBC reports with AVP information that is equivalent to the VSA information below.

The SBC generates SMS call records when you configure the **generate-event** parameter with the **messages** value. Fields supporting message accounting include:

- Acme-Access-Network-Information (248)
- Acme-Visited-Network-Identifier (251)
- Acme-Originating-IOI (249, extension 3)
- Acme-Terminating-IOI (249, extension 4)
- Acme-IMSI (252)
- Acme-IMEI (249, extension 5)
- Acme-Node-Functionality (249, extension 6)
- Acme-SMS Message Type (249, extension 7)
- Acme-SMS Calling party number (249, extension 8)
- Acme-SMS Called party number (249, extension 9)
- Acme-Message Length (249, extension 10)

- Acme-Timestamp

The SBC generates VoLTE call records under the same scenarios and using the same configuration as other SIP calls. Fields supporting VoLTE call accounting include:

- Acme-Access-Network-Information (248)
- Acme-Visited-Network-Identifier (251)
- Acme-Originating-IOI (249, extension 3)
- Acme-Terminating-IOI (249, extension 4)
- Acme-IMSI (252)
- Acme-IMEI (249, extension 5)
- Acme-History-Info (250)
- Acme-Node-Functionality (249, extension 6)
- Acme-P-GW IP Address (249, extension 1)
- Acme-S-GW IP Address (249, extension 2)

Note

The SBC includes this same information within equivalent records managed over diameter. VSAs do not have the "Acme" prefix in their name, and the VSA identification information is specific to diameter VSAs.

Distinct VoLTE Processes

For VoLTE calls, the process for generating CDRs is the largely the same as for other calls. As described, there are additional data points included for these call types.

In addition, the list below presents additional processes reserved for VoLTE data management with which you should be familiar:

- When there is an SRVCC event, the SBC creates a separate set of CDRs for the handover session. The SBC correlates the original and handover session using the "Generic-ID" field, which points to the Call-ID of the initial session. In addition, the SBC populates the Generic-ID field within the Initial Session CDRs (STOP), with the HO session Call-ID.
- The SBC copies the Call id of the second INVITE (Handover INVITE) into the Generic Id into the CDR for the first INVITE (initial call) for both MO and MT call
 - For mobile originating call—When the SBC receives the 200 Ok for the BYE from UE, it inserts the Call id of second INVITE, which is generated from the MSC-S as Generic Id, into the CDR of First MO Invite (Before the handover call).
 - For mobile terminating call—When the SBC receives the 200 Ok for the BYE from UE, it inserts the Call id of the second INVITE, which is generated from the MSC-S as Generic Id, into the CDR of the first MT INVITE (before the handover call).
 - If there is a negative case, such as a BYE timeout, the SBC writes the Call id of second INVITE, which is generated from the MSC-S as the Generic Id, into the CDR of the first INVITE (before the handover call) when that corresponding call gets terminated.

Note

The SBC performs these same processes for both RADIUS accounting when generating CDRs and Diameter accounting when generating ACRs.

Configurations to Specify VoLTE and SMS Data

You can configure the SBC to use specific data in call data records provided over RADIUS, Diameter and within local CSVs. This ensures that the specified fields

There are two configurations available for specifying VoLTE and SMS data:

- Subscribe to IP-CAN-CHANGE events
- Specify Inter-Operator Identifier (IOI)

Subscribe to IP-CAN-CHANGE Events

You can configure a subscription to the IP-CAN-CHANGE event using the Rx interface during the AAA/RAR exchange. To do this, you configure the **ip-can-change** value to the **specific-action-subscription** of the applicable **ext-policy-config**. This causes the SBC to apply the value in the AN-GW-Address AVP from the PCRF as the S-GW IP address.

Note

If the SBC receives more than one AN-GW-Address AVP from the PCRF, it applies the value in the AN-GW-Address AVP from the PCRF. It also uses the S-GW IP address the first AVP as the S-GW IP address.

Option to Specify IOI

You configure the **realm-as-ioi** option in the applicable **account-config** to send the realm name as the IOI in diameter ACRs. If this option is not set, the SBC uses the IOI from the charging vector.

Configure this option using the syntax below.

```
ORACLE(account-config)# options +realm-as-ioi
```

If you type options and then the option value without the plus sign, you overwrite any previously configured options. To add a new option to an options list, pre-pend the new option with a plus sign as shown in the previous example.

Configuring the ip-can-change Subscription

You use the steps below to Subscribe to IP-CAN-CHANGE events at the PCRF and apply the value in the AN-GW-Address AVP from the PCRF as the S-GW IP address.

To obtain the S-GW IP address for mobile originating and terminating scenarios and provide that data in CDRs for RADIUS, diameter and local CSVs:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ORACLE# configure terminal
```

2. Type **media-manager** and press Enter.

```
ORACLE(configure)# media-manager
```

3. Type **ext-policy-config** and press Enter.

```
ORACLE(session-router)# ext-policy-config  
ORACLE(ext-policy-config)#
```

4. Type **specific-action-subscription, ip-can-change** and press Enter.

```
ORACLE(session-router)# specific-action-subscription ip-can-change  
ORACLE(ext-policy-config)#
```

The **specific-action-subscription** accepts multiple values. When configuring 2 or more specific actions, enclose them in quotation marks, with the values separated by spaces.

5. Save your work.

Including P-Visited Network Identifier and History-Info Headers in CDRs

You can configure the SBC to add fully compliant P-Visited Network Identifier (PVNI) and History-Info (HI) headers in CDRs. You configure this by adding the **pcscf-cdr-compliance** option to the **account-config**, specifying whether you want to include PVNI (**PVNI-pref**), HI (**HI-pref**), or both. The behavior is dependent on the type of call, including Mobile Terminating (MT) and Mobile Originating (MO), information provided by SIP, and whether you are also using an S8HR profile.

The PVNI and HI fields in CDRs may or may not contain data. When configured, the SBC performs processes to determine whether or not to add:

- P-Visited-Network-ID to the applicable CDR field
- History-Info to the applicable CDR field

You configure the **pcscf-cdr-compliance** in the applicable **account-config** to use these processes within your environment.

```
ORACLE# configure terminal  
ORACLE(configure)# session-router  
ORACLE(session-router)# account-config  
ORACLE(account-config)# select  
ORACLE(account-config)# options +pcscf-cdr-compliance=PVNI-pref
```

If you save and activate this configuration, the SBC enables PVNI CDR population for MT calls. To configure for both PVNI and HI headers, configure the option with both values separated by a comma and enclosed in quotes.

```
ORACLE(account-config)# options +pcscf-cdr-compliance="PVNI-pref,HI-pref"
```

Support for P-Visited-Network-ID Field

For MT calls, the access SBC, deployed as an A-SBC, inserts the PVNI header in CDRs based on the called party registration cache entry (MCC/MNC). If the Called party registration cache does not have a PVNI value, the A-SBC inserts the **network-id** value from the access side (egress realm) **sip-interface** configuration as the PVNI into CDRs.

For both MO and MT and when you configure it to add PVNI to CDRs, the A-SBC checks for an **s8hr-profile** in the same interface:

- If there is an S8HR profile on the access **sip-interface**:
 - If the SBC receives an MCC/MNC from the Rx server, it creates the PVNI header using the called party registration cache entry (MCC/MNC) and adds it to the CDR.
 - If the SBC does not receive an MCC/MNC, It checks whether there is a **network-id** value on the access side **sip-interface**:
 1. If so, the SBC creates the PVNI using that **network-id** value.
 2. If not, the SBC uses the **local-mccmnc** value as the PVNI, and adds it to CDR.

Note

If you have not configured the **local-mccmnc** value in your S8HR profile, the SBC uses the default, which is 999999.

- If there is not an S8HR profile on the access **sip-interface**, the SBC checks whether there is a **network-id** value on the access side **sip-interface**. If so, the SBC uses the **network-id** value as the PVNI, and adds it to CDR.
- If both the S8HR and the egress (access) **network-id** are not configured, the SBC checks whether the initial INVITE/MESSAGE comes from a trusted endpoint and contains a PVNI:
 - If so, the SBC relays the PVNI and add to CDR.
 - If not, the SBC leaves the PVNI field empty.

When you have set the **pcscf-cdr-compliance** option to include PVNI, and the SBC is acting as an I-SBC handling MO/MT calls, the SBC uses the following sequence for populating the CDR field:

1. If configured, uses the **network-id** on the ingress **sip-interface** as PVNI.
2. If populated and from a trusted endpoint, uses the PVNI from the initial INVITE.
3. Leaves the PVNI field empty.

Note

This behavior applies to the INVITE or any Re-INVITE.

Support for History-Info Field

For MO calls, if you have configured the HI option in the **account-config**, SBC uses the History-Info(s) received in the initial INVITE replies, including those with 181, 180 or 200 status-codes. The SBC populates the CDR with the last provisional (>100) or final (200) response containing History-Info(s). If History-Info is not available in provisional or final replies, the SBC leaves the History-Info in the CDR empty.

For MT calls, SBC extracts History-Info header(s) from the initial INVITE and adds them to the CDR. If History-Info is not available in the initial INVITE, the SBC leaves the HI field empty.

If there are multiple History-Info headers in the initial INVITE, the SBC concatenates all the history-info headers values, and without exceeding the default or configured CDR field size, adds them to the CDR.

For example, assume INVITE has three History-Info headers in the following order:

1. HI-1 - 100 characters
2. HI-2 - 100 characters
3. HI-3 - 100 characters

By default, the maximum CDR field size is 246. In this case, the SBC includes the first two History-Info headers in their entirety, and truncates HI-3.

Consider the presence of the following HI headers:

- History-Info: <sip:bob@example.com>;index=1
- History-Info: <sip:office@example.com>;index=1.2;mp=1
- History-Info: <sip:office@192.0.2.5>;index=1.2.1;rc=1.2

The SBC populates the History-Info CDR as follows

```
<sip:bob@example.com>;index=1, <sip:office@example.com>;index=1.2;mp=1,
<sip:office@192.0.2.5>;index= 1.2.1;rc=1.2
```

The History-Info2 CDR Field

In addition to the History-Info CDR field, the SBC supports the History-Info2 CDR field to capture call history information. The system populates this field when the number of characters for population exceeds the maximum number of characters supported by the History-Info field. If multiple history-info headers generate text that exceeds 246 characters, the SBC parses the headers and adds any spillover into History-Info2. This extends the maximum number of these characters to 492.

For example, if the system receives the following 6 HI headers:

- History-Info: [sip:+34606550955@10.197.141.69;user=phone?Privacy=none];index=1
- History-Info: [sip:unknown@unknown.invalid;cause=404];index=1.1
- History-Info: [sip:unknown@unknown.invalid;cause=404];index=1.1.1
- History-Info: [sip:unknown@unknown.invalid;cause=404];index=1.1.1.1
- History-Info: [sip:+34606550955@10.197.141.69;user=phone;cause=404?Privacy=none];index=1.1.1.1.1
- History-Info: [sip:+34818281924925024@10.197.141.69;user=phone;cause=404;index=1.1.1.1.1.1]

The system includes the first four HI headers in the History Info attribute, and the last two in the History Info2 attribute.

The maximum number of characters for retaining History-Info data is 492 characters. Should your deployment exceed that number of characters, you can consider setting the **cdr-attr-size-limit** option in the **sip-config** to enable the system to drop the earliest History Info headers in the list and avoid truncating the last entries.

P-Asserted-ID Header Format in CDRs

The P-Asserted-ID (PAI) Header URI in CDR displays without the display name by default, even though PAI header is present along with the display-name in the SIP message. The format includes <> (angle brackets) symbol in the Acme-P-Asserted-ID. For example, Acme-P-Asserted-ID = <sip:office@example.com>. To include angle brackets symbol in the Acme-P-

Asserted-ID, add an option **display-name-AVP-add** under **account-config**, so that PAI header URI in CDR, displays along with the display-name if present.

Adding the display-name-AVP-add option

To add an option **display-name-AVP-add** under **account-config** from CLI:

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# account-config
ORACLE(account-config)# options +display-name-AVP-add
```

If you type the option without the plus sign, you overwrite any previously configured options. To append the new option to the options list, prepend the new option with a plus sign as shown in the previous example.

Oracle RADIUS Acme-Extended-Attributes VSAs

As new attributes become available in the RADIUS dictionary, they are accessible on the Acme-Extended-Attributes attribute. These statistics are also available in local CSVs.

MSRP Attributes

MSRP attributes will look like the following when put on one AVP:

```
msrp-attr-type = 0
msrp-attr-len = ...
msrp-attr-data = {1, Acme-MSRP-Calling-Packets, Acme-MSRP-Calling-octets,
Acme-MSRP-Calling-Packets-Transmitted, Acme-MSRP-Calling-Octets-Transmitted,
Acme-MSRP-Called-Packets, Acme-MSRP-Called-Octets, Acme-MSRP-Called-Packets-
transmitted, Acme-MSRP-Called-Octets-Transmitted}
```

For example, if the value for a call is `is0x00230100000001000000f6000000010000009a000000010000009a00000001000000f5`, you should create a parser to break up the information as follows:

- Type: 0x00
- Length: 23
- Version: 01
- Calling Packets: 00000001
- Calling Octets: 000000f6
- Calling Packets Transferred: 00000001
- Calling Octets Transferred: 0000009a
- Called Octets: 0000009a
- Called Packets Transferred: 00000001
- Called Octets Transferred: 000000f5

Table 5-1 MSRP Attributes in CDRs

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------------------|---|-----------------|----------------------|----------|
| Acme-MSRP-Calling-Packets-Received | MSRP total calling (initiating side) packets received by the SBC | 249, extended | integer | Stop |
| Acme-MSRP-Calling-Octets-Received | MSRP total calling (initiating side) octets received by the SBC | 249, extended | integer | Stop |
| Acme-MSRP-Calling-Packets-Transmitted | MSRP total calling (initiating side) transmitted packets by the SBC | 249, extended | integer | Stop |
| Acme-MSRP-Calling-Octets-Transmitted | MSRP total calling (initiating side) transmitted octets by the SBC | 249, extended | integer | Stop |
| Acme-MSRP-Called-Packets-Received | MSRP total called (answering side) received packets by the SBC | 249, extended | integer | Stop |
| Acme-MSRP-Called-Octets-Received | MSRP total called (answering side) octets received by the SBC | 249, extended | integer | Stop |
| Acme-MSRP-Called-Packets-Transmitted | MSRP total called (answering side) transmitted packets by the SBC | 249, extended | integer | Stop |
| Acme-MSRP-Called-Octets-Transmitted | MSRP total called (answering side) transmitted octets by the SBC | 249, extended | integer | Stop |

STIR/SHAKEN Attributes

This table lists and describes the STIR/SHAKEN attributes used for RADIUS CDRs. The table includes attribute name, attribute description, attribute value, attribute value type, and the list of CDR message types for which the value is captured.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|--------------------------|--|-------------------|----------------------|--------------------------|
| Acme-Stir-Signed-Request | This AVP indicates that the SBC has sent the signing request to the STI-AS and received a response. Values for this AVP mean: <ul style="list-style-type: none"> • 1— The SBC has signed the INVITE. • 0—The SBC has not signed the INVITE. • 2—Signing is not applicable for the INVITE. | 249, extension 11 | String | START INTERIM STOP |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---|---|----------------------|----------------------|--------------------------|
| Acme-Stir-Signed-Request-Exception-Id | This AVP indicates that the SBC did not sign the INVITE because of a failure response from the STI-AS. It includes the ATIS defined service and policy exception id (e.g. SVC4000, SVC4001, POL4050). | 249, extension 12 | String | START INTERIM STOP |
| Acme-Stir-Verification-Request | This AVP indicates that the SBC has sent the verification request to the STI-VS and received a response. Values for this AVP mean: <ul style="list-style-type: none"> 1— The SBC has verified the request. 0—The SBC has not signed verified the request. 2—Request verification is not applicable for the call. | 249, extension 13 | String | START INTERIM STOP |
| Acme-Stir-Verification-Request-Exception-Id | This AVP indicates that the SBC did not verify the request because of a failure response from the STI-VS. It includes the ATIS defined service and policy exception id (e.g. SVC4000, SVC4001, POL4050). | 249, extension 14 | String | START INTERIM STOP |
| Stir-VS-Verstat | The SBC populates this AVP with the verstat values the SBC includes in the SIP INVITE following the Stir Shaken trigger. Potential values include TN-Validation-Passed, TN-Validation-Failed or No-TN-Validation | 249, extension 15 | String | START INTERIM STOP |
| Stir-VS-Reason | The SBC populates this AVP with received reason information. | 249, extension 16 | String | START INTERIM STOP |
| Stir-TN-Used-For-AS-VS-Request | This AVP contains the TN number captured by the SBC. TN selection uses the following priority: <ol style="list-style-type: none"> 1. From the Tel PAI (if present) 2. From the SIP PAI (if present, and Tel PAI is not present) 3. From the "From" header if both Tel and SIP PAI are not present. | 249, extension 17 | String | START INTERIM STOP |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---------------------------|--|-------------------|----------------------|--------------------------|
| Stir-Div-Signed-Request | Upon sending the DIV signing request to the STI-AS and receiving a 200 OK successful response from STIR-AS, the SBC enumerates the results in this AVP as follows: <ul style="list-style-type: none"> • 1—indicates DIV signing the INVITE • 0—indicates no DIV signing the INVITE • 2—indicates DIV signing is not applicable on the INVITE | 249, extension 18 | String | START INTERIM STOP |
| Stir-Div-Verified-Request | Upon sending the DIV verify request to the STI-AS and receiving a 200 OK successful response from STIR-VS, the SBC enumerates the results in this AVP as follows: <ul style="list-style-type: none"> • 1—indicates DIV verification the INVITE • 0— indicates no DIV verification the INVITE • 2—indicates DIV verification is not applicable on the INVITE | 249, extension 19 | String | START INTERIM STOP |
| Stir-VS-Invite-State | Indicated SBC action on call based on STIRVS rejection configuration. <ul style="list-style-type: none"> • Continued • Terminated | 249, extension 20 | String | START INTERIM STOP |

STIR/SHAKEN attributes are embedded under one extended Radius AVP.

- type = 26 (vendor-specific)
- length = ...
- vendor-id = 9148 (acme)
- vendor-type = 249 (extended-attributes)
- vendor-length = ...
- vendor-data = { x-attr-type (16-bit), x-attr-len (16-bit), x-attr-data (variable) }

An example of an AVP presenting the Acme-Stir-Signed-Request attribute indicating the SBC has signed the INVITE appears as follows:

```
x-attr-type = 1 (Acme-Stir-Signed-Request) [1 byte]
x-attr-len = ... [1 byte]
x-attr-data = { version[1 byte], Acme-Stir-Signed-Request [depending on length]}
```

Additional STIR/SHAKEN Attributes

The table below specifies additional STIR/SHAKEN Oracle VSAs used for RADIUS verification. These VSAs are not extensions of Oracle attribute value 249.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|--|--|-----------------|----------------------|--------------------------|
| Stir-VS-attest Note: The system populates this AVP when MAN compliance is enabled. | Populate the attest claim value from the STI-VS verification response within the JSON body for 3GPP deployments. If there is no attest claim in provided by the verification response, the system populates the Sti-VS-Attest field with empty values in all CDRs. Values include A or B or C | 178 | String | START INTERIM STOP |
| Stir-Div-VS-Reason Note: The system populates this AVP when MAN compliance is enabled. | This object collects the reason value with the concatenation of the "reasoncode" and "reasontext" obtained from each Div Verify result of JSON body. The system collects the reasoncode and reasontext for Div Verify result, even if they are same. Example: SIP ; cause=438 ;text="Identitysignaturepayloaddoesnotmatch here-constructedone", ;cause=410 ;text="Application inactive" | 179 | String | START INTERIM STOP |

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|---|--|-----------------|----------------------|--------------------------|
| Stir-Div-VS-Verstat Note: The system populates this AVP when MAN compliance is enabled. | This object collects the verstat values, enclosed in double quotes, for each verified div PASSporT. The system collects the value for each DIV verification result, even if they use the same code. Examples include: "TN-Validation-Passed, TN-Validation-Failed, TN-Validation-Passed" | 180 | String | START INTERIM STOP |
| Stir-AS-Bypass-Token | This object contains the static bypass token added to the P-Identity-Bypass header in the egress INVITE for calls that bypassed the STI-AS process due to matching a configured bypass scenario. | 181 | String | START INTERIM STOP |

Modem Tone Attributes

The following table describes the modem tone attribute, embedded under the Acme-Extended-Attributes-1 AVP.

| Attribute Name | Attribute Description | Attribute Value | Attribute Value Type | Messages |
|----------------------------|---|-----------------|----------------------|-----------------|
| Acme-Extended-Attributes-1 | Indicates VBD modem tones that were detected on the call. | 182 | String | INTERIM STOP |

The attribute takes the following form:

```

type           = 26 (vendor-specific)
length        = ...
vendor-id     = 9148 (acme)
--VSA--
vendor-type   = 182 (Media-Interworking-Fields)
vendor-length = ...
vendor-data   = { x-attr-type (1 byte), x-attr-len (1 byte), x-attr-data
(version[1 byte], data) }

```

For example, if the value for a call is 0x0713014d6f64656d2d416e732d2f414e53616d, you should create a parser to break up the information as follows:

- Type: 0x07
- Length: 13
- Version: 01
- VBD Tone Name: 4d6f64656d2d416e732d2f414e53616d (Modem-Ans-/ANSam)

Supported modem tones are:

- Modem-Ans-ANS
- Modem-Ans-/ANS
- Modem-Ans-ANSam
- Modem-Ans-/ANSam
- Modem-Ans-Bell103
- Modem-Ans-V21High
- Modem-Ans-V23High
- Modem-Ans-V22
- Modem-Ans-V22+1800Hz
- Modem-Ans-V8bis
- Modem-Ans-Baudot
- Modem-Orig-Bell103
- Modem-Orig-V21Low
- Modem-Orig-V25Calling
- Modem-Orig-V8bis
- Modem-Orig-Baudot

Mappings and Disconnect Cause Values

This section provides information about H.323 and SIP disconnect cause values for RADIUS CDRs generated by the SBC.

SIP H.323 and Q.850 Mappings

This section provides tables that show the mappings between SIP Status and: H.323 Disconnect Reason, H.323 Release Complete Reason, and RAS error. It also shows the mapping for Q.850 cause to H.323 Release Complete Reason.

SIP Status to H.323 Disconnect Reason Mapping

| SIP Status | H.323 Disconnect Reason |
|-----------------------------|-------------------------|
| 480 Temporarily Unavailable | No Bandwidth |
| 404 Not Found | Gatekeeper Resource |
| 404 Not Found | Unreachable Destination |
| 603 Decline | Destination Rejection |

| SIP Status | H.323 Disconnect Reason |
|-----------------------------|--------------------------|
| 505 Version Not Supported | Invalid Revision |
| 401 Unauthorized | No Permission |
| 503 Service Unavailable | Unreachable Gatekeeper |
| 480 Temporarily Unavailable | Gateway Resource |
| 400 Bad Request | Bad Format Request |
| 486 Busy Here | Adaptive Busy |
| 486 Busy Here | In Conference |
| 500 Internal Server Error | Undefined Reason |
| 486 Busy Here | Facility Call Deflection |
| 401 Unauthorized | Security Denied |

SIP Status to H.323 RAS Error Mapping

| SIP Status | H.323 RAS Error |
|---------------------------|---------------------------|
| 404 Not Found | Gatekeeper Resource |
| 401 Unauthorized | Invalid Permission |
| 503 Service Unavailable | Request Denied |
| 500 Internal Server Error | Undefined |
| 401 Unauthorized | Caller Not Registered |
| 305 User Proxy | Route Call to Gatekeeper |
| 500 Internal Server Error | Invalid Endpoint ID |
| 503 Service Unavailable | Resource Unavailable |
| 401 Unauthorized | Security Denial |
| 501 Not Implemented | QoS Control Not Supported |
| 484 Address Incomplete | Incomplete Address |
| 302 Moved Temporarily | Route Call to SCN |
| 485 Ambiguous | Aliases Inconsistent |
| 401 Unauthorized | Not Currently Registered |

SIP Status to H.323 Release Complete Reason Error Mapping

| SIP Status | H.323 RAS Error |
|-----------------------------------|-------------------------|
| 300 Multiple Choices | Undefined Reason |
| 401 Unauthorized | Security Denied |
| 402 Payment Required | Undefined Reason |
| 403 Forbidden | No Permission |
| 404 Not Found | Unreachable Destination |
| 405 Method Not Allowed | Undefined Reason |
| 606 Not Acceptable | Undefined Reason |
| 407 Proxy Authentication Required | Security Denied |
| 408 Request Timeout | Adaptive Busy |
| 409 Conflict | Undefined Reason |
| 410 Gone | Unreachable Destination |
| 411 Length Required | Undefined Reason |
| 414 Request-URI Too Large | Bad Format Address |

| SIP Status | H.323 RAS Error |
|-------------------------------------|--------------------|
| 415 Unsupported Media Type | Undefined Reason |
| 420 Bad Extension | Bad Format Address |
| 480 Temporarily Unavailable | Adaptive Busy |
| 481 Call/Transaction Does Not Exist | Undefined Reason |
| 482 Loop Detected | Undefined Reason |
| 483 Too Many Hops | Undefined Reason |
| 484 Address Incomplete | Bad Format Address |

Q.850 Cause to H.323 Release Complete Reason Mapping

The table below describes how the Q.850 Causes and the H.323 release complete reasons are mapped internally on the SBC.

| Q.850 Cause | Numeric Code | H.323 Release Complete Reason |
|--------------------------------|--------------|-------------------------------|
| Not Route To Destination | 3 | Unreachable Destination |
| Normal Call Clearing | 16 | Destination Rejection |
| User Busy | 17 | In Conference |
| Subscriber Absent | 20 | Called Party Not Registered |
| Invalid Number Format | 28 | Bad Format Address |
| Normal Unspecified | 16 | Undefined Reason |
| No Circuit/Channel Available | 34 | No Bandwidth |
| Network Out of Order | 38 | Unreachable Gatekeeper |
| Temporary Failure | 41 | Adaptive Busy |
| Switching Equipment Congestion | 42 | Gateway Resource |
| Resource Unavailable | 47 | Gatekeeper Resource |
| Incompatible Destination | 88 | Invalid Revision |
| Interworking Unspecified | 111 | No Permission |

SIP-SIP Calls

The SBC maps SIP status codes and events to disconnect cause attribute values used by Cisco Systems Proxy Server (CSPS) accounting services.

| SIP Status Category/Event | CDR Disconnect Cause | Description |
|---------------------------|----------------------|---------------------|
| Undetermined reason | 0 | Undetermined reason |
| BYE | 1 | Normal clearing |
| 3xx: Redirection | 2 | Redirection |
| 4xx: Client Error | 3 | Client error |
| 5xx: Server Error | 4 | Server error |
| 6xx: Global Failure | 5 | Global error |

SIP-H.323 Calls with Interworking

For calls that require SIP-H.323 interworking, the SBC generates two sets of RADIUS CDRs: one for the SIP call-leg and one for the H.323 call leg. The values recorded in RADIUS Stop

records for the disconnect cause depend on the nature and source of the call disconnect or rejection.

SIP Events and Errors

For calls rejected or disconnected because of SIP events and errors, the SBC records Q.850 cause values mapped from the SIP event/status code in the SIP CDR. For the H.323 CDR, the SIP status categories and events are mapped to Q.850 cause codes.

The entries in this table are determined by the [SIP Status to H.323 Release Complete Reason Error Mapping](#).

| SIP Status Category/Event | SIP CDR Disconnect Cause | H.323 Disconnect Cause Value (Q.850) |
|-----------------------------------|-----------------------------------|--------------------------------------|
| BYE | 16—Normal call clearing | 16—Normal call clearing |
| 3xx | 23—Redirection to new destination | 16—Normal call clearing |
| 404 Not Found | 21—Call rejected | 3—No route to destination |
| 410 Gone | 21—Call rejected | 3—No route to destination |
| 403 Forbidden | 21—Call rejected | 111—Interworking unspecified |
| 408 Request Timeout | 21—Call rejected | 41—Temporary failure |
| 413 Request Entity Too Big | 21—Call rejected | 28—Invalid number format |
| 414 Request URI Too Large | 21—Call rejected | 28—Invalid number format |
| 420 Bad Extension | 21—Call rejected | 28—Invalid number format |
| 484 Address Incomplete | 21—Call rejected | 28—Invalid number format |
| 408 Request Timeout | 21—Call rejected | 41—Temporary failure |
| 480 Temporarily unavailable | 21—Call rejected | 41—Temporary failure |
| 486 Busy Here | 21—Call rejected | 17—User Busy |
| 401 Unauthorized | 21—Call rejected | 32—Normal unspecified |
| 407 Proxy Authentication Required | 21—Call rejected | 32—Normal unspecified |
| All other 4xx | 21—Call rejected | 16—Normal unspecified |
| 502 Bad Gateway | 38—Network out of order | 28—Invalid number format |
| 505 Bad Version | 38—Network out of order | 88—Incompatible destination |
| All other 5xx | 38—Network out of order | 16—Normal unspecified |
| 600 Busy Everywhere | 31—Normal unspecified | 41—Temporary failure |
| 603 Decline | 31—Normal unspecified | 31—Normal unspecified |
| 604 Does Not Exist Anywhere | 31—Normal unspecified | 3—No route to destination |
| All other 6xx | 31—Normal unspecified | 31—Normal unspecified |

H.323 Events and Errors

The Q.850 cause code value is recorded for the disconnect cause in the CDR for the H.323 call leg if the Q.850 cause is received. H.323 recommendations state that either Q.850 Cause of RelCompReason is mandatory for the RELEASE COMPLETE; the Cause information element (IE) is optional everywhere. The Cause IE and the ReleaseCompleteReason (part of the release complete message) are mutually exclusive.

If a Q.850 cause code is not received, the SBC records a Q.850 cause value mapped from the received ReleaseCompleteReason as defined in the table below.

The entries in this table are determined by the [SIP Status to H.323 Disconnect Reason Mapping](#).

| H.323 ReleaseCompleteReason | H.323 CDR Disconnect Cause | SIP Status | SIP CDR Disconnect Cause |
|-----------------------------|-----------------------------------|-----------------------------|--------------------------|
| No Bandwidth | 34—No channel/circuit available | 480 Temporarily Unavailable | 21—Call rejected |
| Gatekeeper Resource | 47—Resource unavailable | 404 Not Found | 21—Call rejected |
| Unreachable Destination | 3—No route to destination | 404 Not Found | 21—Call rejected |
| Destination Rejected | 31—Normal unspecified | 603 Decline | 31—Normal unspecified |
| Invalid Revision | 88—Incompatible destination | 505 Version Not Supported | 38—Network out of order |
| No Permission | 111—Interworking unspecified | 401 Unauthorized | 21—Call rejected |
| Unreachable Gatekeeper | 38—Network out of order | 503 Service Unavailable | 38—Network out of order |
| Gateway Resource | 42—Switching equipment congestion | 480 Temporarily unavailable | 21—Call rejected |
| Bad Format Request | 28—Invalid number format | 400 Bad request | 21—Call rejected |
| Adaptive Busy | 41—Temporary failure | 486 Busy Here | 21—Call rejected |
| In Conference | 17—User busy | 486 Busy Here | 21—Call rejected |
| Undefined Reason | 16—Normal unspecified | 500 Internal Server Error | 38—Network out of order |
| Called Party Not Registered | 20—Subscriber absent | 404 Not Found | 21—Call rejected |
| Caller Not Registered | 31—Normal call clearing | | |
| New Connection Needed | 47—Resource Unavailable | 401 Unauthorized | 21—Call rejected |

H.225 RAS Errors

For calls that are rejected because of H.225 RAS, there is no CDR generated for the H.323 call leg as no Setup message is generated. The SBC maps RAS errors to SIP Status as specified in the table below. The SIP CDR disconnect cause values are the same as the CSPS disconnect cause values already mentioned and defined.

The entries in this table are determined by the [SIP Status to H.323 RAS Error Mapping](#).

| H.225 RAS Error | SIP Status | SIP CDR Disconnect Cause |
|-----------------------------|---------------------------|-----------------------------------|
| Called Party Not Registered | 404 Not Found | 21—Call Rejected |
| Invalid Permission | 401 Unauthorized | 21—Call Rejected |
| Request Denied | 503 Service Unavailable | 38—Network out of order |
| Undefined | 500 Internal Server Error | 38—Network out of order |
| Caller Not Registered | 401 Unauthorized | 21—Call Rejected |
| Route Call to Gatekeeper | 305 Use Proxy | 23—Redirection to new destination |
| Invalid Endpoint ID | 500 Internal Server Error | 38—Network out of order |
| Resource Unavailable | 503 Service Unavailable | 38—Network out of order |
| Security Denial | 401 Unauthorized | 21—Call Rejected |
| QoS Control Not Supported | 501 Not Implemented | 38—Network out of order |
| Incomplete Address | 484 Address Incomplete | 21—Call Rejected |
| Route Call to SCN | 302 Moved Temporarily | 2—Redirection |

| H.225 RAS Error | SIP Status | SIP CDR Disconnect Cause |
|--------------------------|-------------------|---------------------------------|
| Aliases Inconsistent | 485 Ambiguous | 21—Call Rejected |
| Not Currently Registered | 401 Unauthorized | 21—Call Rejected |

6

Diameter Accounting

The SBC supports the Diameter charging interface, Rf. This interface provides similar functionality to the RADIUS interface, but utilizes Diameter as the underlying application layer protocol. As a result, the SBC can integrate more thoroughly with IMS standards as well as provide a more dynamic, secure, and robust accounting interface.

Diameter Accounting Messages

The Rf interface can send messages based on the signaling application's actions. These messages are Accounting Charging Request (ACR) Start messages, ACR Stop messages, Event messages and Interim messages.

Resending ACRs

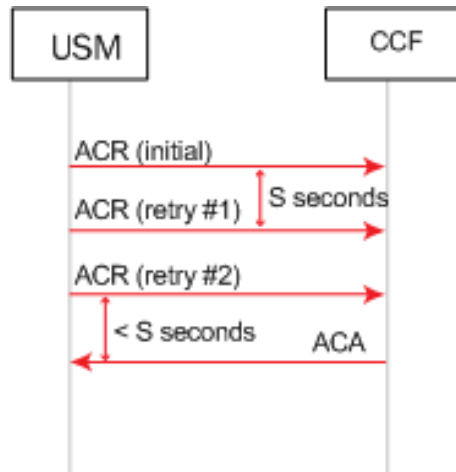
- If an ACA is not received to acknowledge the reception of an ACR, the SBC attempts to resend an ACR and buffers all subsequent ACRs for the same session until the acknowledgement is received. Once the acknowledgement is received from the CCF, all buffered ACRs for that same session may be sent to the CCF in the appropriate order. If the SBC does not receive the ACA after the user-specified number of retries, then the SBC sends all of the buffered ACR records for a session to the secondary CCF. (The number of ACR retries as well as the wait time in between retries is configurable by using the max-acr-retries account configuration parameters and acr-retry-interval, respectively.)

Postponement Feature

- Any number of ACRs can be sent during a session, including Start, Stop, Interim and ACR messages. The ACR postponement feature (non-configurable) ensures that the next ACR is not sent until the previous ACR is acknowledged with an ACA.

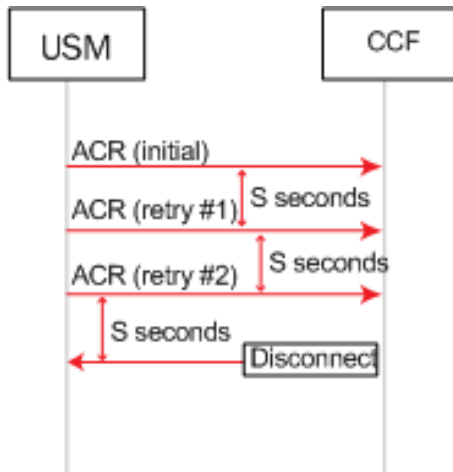
Call Flow Examples

- The following call flow example shows success in receiving an ACA for a session after resending the ACR message three times.



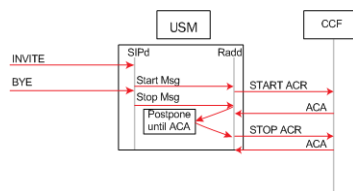
Successful ACA Acknowledgement After Three Retries

The following call flow example shows the failure to receive an ACA for a session after sending the ACR message three times.



Unsuccessful ACA Acknowledgement After Three Retries

The following call flow example shows how the delay of the ACA acknowledgement for a session results in a postponement until the ACA is finally received. During the postponement, the ACRs are buffered until the corresponding ACA is received. If an ACR for one session is postponed, it does not delay the other session's ACRs.



Call Flow Showing Delivery Postponement

- Additional ACR Interim messages are sent when service changes; this roughly maps to a RADIUS Interim-Update message. See [Accounting-Record-Type AVP \(480\)](#).
- ACR Stop messages are sent at the end of service delivery.

The SBC sends a set of AVPs in each ACR start and stop message that make up the charging data. The following table lists which AVPs are included in ACR Start and ACR Stop messages. Individual AVP descriptions are located in the following section.

| AVP | ACR Start | ACR Stop |
|-------------------------------------|-----------|----------|
| Session-Id AVP (263) | X | X |
| Origin-Host AVP (264) | X | X |
| Origin-Realm AVP (296) | X | X |
| Destination-Realm AVP (283) | X | X |
| Destination-Host AVP (293) | X | X |
| Accounting-Record-Type AVP (480) | X | X |
| Accounting-Record-Number AVP (485) | X | X |
| Acct-Application-Id AVP (259) | X | X |
| User-Name AVP (1) | X | X |
| Event-Timestamp AVP (55) | X | X |
| Event-Type AVP (823) | X | X |
| SIP-Method AVP (824) | | |
| Content-Type AVP (826) | | |
| Content-Length AVP (827) | | |
| Role-of-Node AVP (829) | X | X |
| User-Session-Id AVP (830) | X | X |
| Calling-Party-Address AVP (831) | X | N/A |
| Called-Party-Address AVP (832) | X | N/A |
| Time-Stamps AVP (833) | X | X |
| SIP-Request-Timestamp AVP (834) | | |
| SIP-Response-Timestamp AVP (835) | | |
| Inter-Operator-Identifier AVP (838) | X | X |
| Originating-IOI AVP (839) | | |
| Terminated-IOI AVP (840) | | |
| SDP-Session-Description AVP (842) | X | N/A |
| Session-Media-Component AVP (845) | X | N/A |
| SDP-Media-Name AVP (844) | | |
| SDP-Media-Description AVP (845) | | |
| Cause AVP (860) | N/A | X |
| Cause-Code AVP (861) | | |
| Node-Functionality AVP (862) | | |

Per Realm Accounting Control

You can enable or disable accounting control for specific ingress realms using the **accounting-enable** parameter. This parameter is enabled by default.

The SBC's SIP and H.323 tasks check whether this parameter is set to enabled or disabled, and sends records on that basis.

Note

This realm configuration does not trigger accounting of MESSAGE and REGISTER traffic.

Egress Realm Accounting Option

You can also set the **force-realm-accounting** option in the **sip-config** to ensure the SBC performs accounting based on this configuration on the egress realm.

For session accounting, the SBC checks the realm accounting configuration on both the ingress and egress realms. If the accounting configuration is enabled on either realm, and you have enabled the **sip-config** option, the SBC performs accounting for that session. If you have not enabled the **sip-config** option, the SBC only performs accounting for calls at the ingress realms that have accounting enabled.

This option also causes the SBC to perform accounting for REGISTER and MESSAGE requests. For this purpose, the SBC first checks whether you have configured the **sip-config** option. If set, the SBC checks the realm accounting configuration on the ingress and egress realms. If you have configured realm accounting configuration on either realm, the SBC performs accounting on these messages.

The syntax for this option follows.

```
ORACLE(sip-config)# options + force-realm-accounting
```

Both the **force-realm-accounting** option and the **accounting-enable** parameter are RTC supported.

ACLI Instructions

To configure per realm accounting:

1. Access the **realm-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# media-manager
ORACLE(media-manager)# realm-config
ORACLE(realm-config)#
```

2. Select the **realm-config** object to edit.

```
ORACLE(realm-config)# select
identifier:
1: realm01 left-left:0 0.0.0.0

selection: 1
ORACLE(realm-config)#
```

3. **accounting-enable**—Either leave this parameter set to enabled (default) to generate CDRs for this realm, or change it to disabled.
4. Type **done** to save your configuration.

ACR AVP Descriptions

This section provides individual AVP descriptions.

Session-Id AVP (263)

Uniquely identifies this session. It is a string value and is delimited by semi-colons. This AVP is created according to the Session-Id AVP (AVP Code 263) specified in IETF RFC 3588. An example of a Session-Id from the SBC is as follows, acmesystem;0;1.

Origin-Host AVP (264)

Contains the account-server configuration element's **hostname** parameter followed by the "@" character, followed by the account-server configuration element's **origin-realm** parameter. For example: acmesystem@wancom.com.

Origin-Realm AVP (296)

Contains the **account server** configuration element's **origin-realm** and **domain-name-suffix** parameters where the server request is sent.

Destination-Realm AVP (283)

Contains the value of the Origin-Realm AVP in the CEA received from the accounting server for this connection.

Destination-Host AVP (293)

Contains the value of the Origin-Host AVP in the CEA received from the accounting server for this connection.

Accounting-Record-Type AVP (480)

Contains the value indicating the type of accounting message being sent.

- event record = 1
- start record = 2
- interim record = 3
- stop record = 4

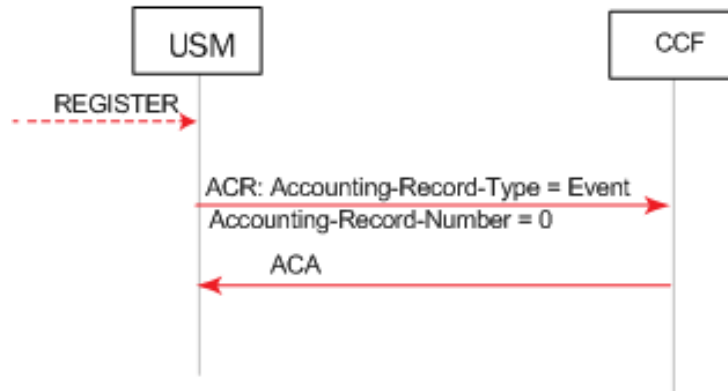
Accounting-Record-Number AVP (485)

A value that uniquely identifies this message in the session (i.e., a sequence number for this connection). The sequence number is assigned sequentially starting with 0 as described below. This is compliant with RFC 3588. The combination of the Accounting-Record-Number AVP and the Session-Id AVP (both of which are unique for the given session) are used to match accounting records with confirmations. This is done by assigning the noted values to the records listed below:

- Event Record — the system assigns this record a value of 0 to this record.
- Start Record — the system assigns this record a value of 0 to this record.
- Interim Record — the system assigns this record a value of 1 to the first record of this type for the session, and increments the value by 1 for each subsequent Interim_record until the value for the Stop_record is more for the last Interim_record for the session.

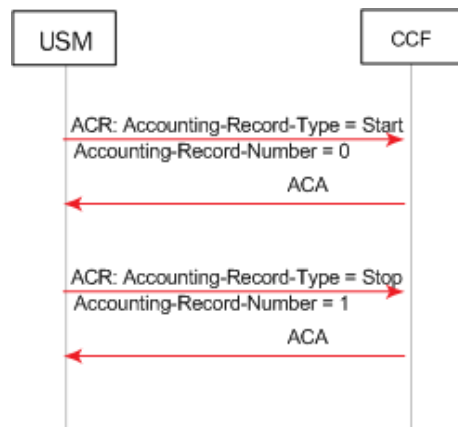
- Stop Record — (see description for Interim_record in the previous bullet) — If there is no Interim_record for the session, the system assigns a value to this record of 1.

The following example call flow shows a Register Event that shows that the Event record in the Accounting-Record-Number AVP is always populated with 0.



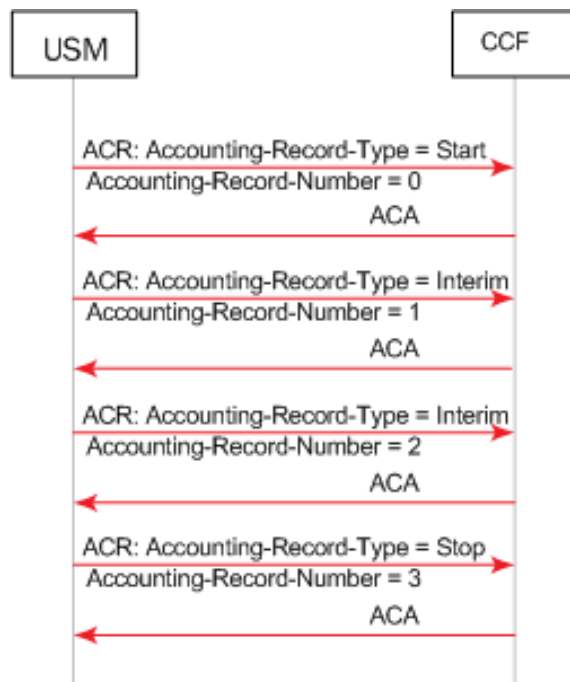
Register Event Call Flow Example

The following example call flow shows session accounting messages with no Interim records. Note that the Start Accounting-Record-Number equals 0 and the Stop Accounting-Record-Number equals 1.



Call Flow Example Showing Session Accounting Messages with No Interim Records

The following example call flow shows session accounting messages with Interim records. Note that the Start record Accounting-Record-Number equals 0 and that the Interim Accounting-Record-Numbers start with a value of 1 and increase by 1 with the second Interim record. The Stop Accounting-Record-Number equals 3.



Call Flow Example Showing Session Accounting Messages with Interim Records

Acct-Application-Id AVP (259)

Set to value "3". This value indicates Diameter-based accounting messages.

User-Name AVP (1)

Contains the account-server configuration element's hostname parameter followed by the "@" character, followed by the account-server configuration element's origin-realm parameter. For example: acmesystem@wancom.com.

Event-Timestamp AVP (55)

Contains the number of seconds since January 1, 1900 when this accounting event took place.

Event-Type AVP (823)

A grouped AVP containing information about the signaling event. It contains the following AVPs:

- SIP-Method AVP (824)—Contains the exact string payload from the SIP request line; i.e., the Method that triggered the accounting event.
- Content-Type AVP (826)—Contains the exact string payload from the "Content-Type" SIP header of the message that triggered the accounting event.
- Content-Length AVP (827)—Contains the exact string payload from the Content-Length" SIP header of the message that triggered the accounting event.

Role-of-Node AVP (829)

Set to the value 2 which indicates that the SBC is operating in a PROXY role.

User-Session-Id AVP (830)

Contains VSA 44 as used in the RADIUS interface.

Calling-Party-Address AVP (831)

The Calling-Party-Address AVP (AVP code 831) is of type UTF8String and holds the address (SIP URI or TEL URI) which identifies the party (Public User Identity or Public Service Identity) initiating a SIP transaction. . It is obtained from the P-Asserted-Identity header of any non-REGISTER SIP Request, either initiating a dialog or a standalone transaction. This AVP may appear several times when the P-Asserted-Identity header contains both a SIP URI and a TEL URI. In case no P-Asserted-Identity is known, this AVP list shall include one item with the value "unknown".

Called-Party-Address AVP (832)

The Called-Party-Address AVP (AVP code 832) is of type UTF8String. In IMS charging (except for SIP Register and SIP Subscription transactions), it holds the address (SIP URI, TEL URI or URN) of the party (Public User ID or Public Service ID) to whom the SIP transaction is posted. The Called Party Address shall be populated with the SIP URI or TEL URI contained in the Request-URI of the outgoing request.

For a registration procedure, this field holds the party (Public User ID) to be registered. In this case, the Called Party Address field is obtained from the To SIP header of the SIP Request. For a subscription procedure this field holds the address of the resource for which the originator wants to receive notifications of change of states. In this case, the Called Party Address field is obtained from the outgoing Request-URI of the SIP Request.

Acme-SipHdr-TO AVP (122)

The AVP is a vendor-specific UTF8 string (D_AVP_ACME_SIP_TO_HDR), which uses the AVP code 122, to capture TO headers from incoming SIP messages and populate the Acme-SipHdr-TO in the Acme-Packet-Specific-Extension-Rf (AVP Code 1) grouped AVP.

This AVP uses the following ABNF grammar.

```
< Acme-SipHdr-TO>:: = < AVP Header: 122 >  
[AVP_ACME_SIP_TO_HDR]
```

By default, the SBC does not populate this AVP. You configure the **sip-tohdr-in-acr** option in the **account-config** to enable this behavior.

Time-Stamps AVP (833)

A grouped AVP that contains timestamps for the related SIP signaling. It contains the following AVPs.

- SIP-Request-Timestamp AVP (834)—A UTC formatted timestamp that corresponds to when the SIP INVITE that started the session was received.
- SIP-Response-Timestamp AVP (835)—A UTC formatted timestamp that corresponds to when the SIP 200 OK response to the INVITE that started the session was received.

Inter-Operator-Identifier AVP (838)

A grouped AVP that indicates the ingress and egress networks from the SBC's perspective. It contains the following AVPs.

- Originating-IOI AVP (839)—The realm where the SBC received the SIP signaling messages.
- Terminated-IOI AVP (840)—The realm where the SIP signaling message exit the SBC.

SDP-Session-Description AVP (842)

This AVP may occur multiple times in an ACR message. It is populated with SDP attribute-lines from the SIP messages to which this ACR Start or Interim message refers. Thus, all "i=", "c=", "b=", "k=", "a=", etc., lines comprise multiple instances of this AVP.

If the SBC is configured to generate Start events on the INVITE, the calling SDP will be used; if the SBC is configured to generate Start events on the OK, the called SDP will be used. ONLY IN ACR Start.

Session-Media-Component AVP (845)

A grouped AVP that contains information about the media session. It contains the following AVPs. ONLY IN ACR Start.

- SDP-Media-Name AVP (844)—populated with the "m=" line from the SDP being used.
- SDP-Media-Description AVP (845)—this AVP may occur multiple times in this grouped AVP. It is populated with SDP attribute-lines from the media component as specified by the media described in the SDP-Media-Name AVP. Thus, all "i=", "c=", "b=", "k=", "a=", etc..., lines related to the above specified "m=" line comprise multiple instances of this AVP.

Cause AVP (860)

A grouped AVP that contains the reason for the termination event and the role/function of the node where the call was terminated. It contains the following AVPs.

- Cause-Code AVP (861)—See Values for Cause Code AVP (861) below.
- Node-Functionality AVP (862)—Set to value 0.

Values for Cause Code AVP (861)

As described in 3GPP TS32.229, the Cause-Code AVP 861 includes the cause code value sent by the IMS node. It is used in stop and/or event messages.

If the session terminated as a result of a specific known SIP error code, the SIP error code is used as the cause code value. Otherwise, cause code values less than or equal to 0 are used for successful causes while values greater than or equal to 1 are used for failure causes.

- Cause code value set to 0 — indicates "Normal end of session" and is used in Accounting-request[stop] message to indicate that an ongoing SIP session has been normally released either by the user or by the network (SIP BYE message initiated by the user or initiated by the network has been received by the IMS node after the reception of the SIP ACK message).

- Cause code value set to "2xx Final Response" (except 200) — used when the SIP transaction is terminated due to an IMS node receiving/initiating a 2xx Final response.
- Cause code value set to "2xx Final Redirection"— used when the SIP transaction is terminated due to an IMS node receiving/initiating a 3xx response.
- Cause code value set to "1"— indicates "Unspecified error" and is used when the SIP transaction is terminated due to an unknown error.
- Cause code value set to "4xx Request failure"— used when the SIP transaction is terminated due to an IMS node receiving/initiating a 4xx error response.
- Cause code value set to "5xx Server failure"— used when the SIP transaction is terminated due to an IMS node receiving/initiating a 5xx error response.
- Cause code value set to "6xx Global failure"— used when the SIP transaction is terminated due to an IMS node receiving/initiating a 6xx error response.
- Cause code value set to "Unsuccessful session setup"— used in the Accounting-request[stop] when the SIP session has not been successfully established (i.e. Timer H expires and SIP ACK is not received or SIP BYE is received after reception of the 200OK final response and SIP ACK is not received).
- Cause code value set to "Internal error"— used when the SIP transaction is terminated due to an IMS node internal error (e.g. error in processing a request/response).

Acme-VBDTone-Detected (162)

Contains the modem tones detected during the session. The modem tones are captured in the following format:

Modem-direction-name

where *direction* is **Ans** or **Orig**, depending on the direction the tone came from and *name* is the name of the tone.

For example, a V.22/V.22bis 2250Hz+1800Hz tone detected on the answer side would be logged as **Modem-Ans-V22+1800Hz**.

STIR/SHAKEN AVPs

The SBC issues custom AVPs that capture STIR/SHAKEN information within ACRs. These are in addition to the AVPs that are common to all traffic.

This table lists STIR/SHAKEN AVPs for DIAMETER CDRs and captures the applicable CDR message types.

| AVP Name | Number | Start | Interim | Stop | Message Type = INVITE | AVP Type |
|----------------------------------|--------|-----------------------------|---------|------|-----------------------|----------|
| Stir-Signed-Request | 104 | Yes (On 200OK of invite) | Yes | Yes | Yes | String |
| Stir-Signed-Request-Exception-Id | 105 | Yes (On 200OK of invite) | Yes | Yes | Yes | String |
| Stir-Verified-Request | 106 | Yes | Yes | Yes | Yes | String |

| AVP Name | Number | Start | Interim | Stop | Message Type = INVITE | AVP Type |
|------------------------------------|--------|------------------------------|---------|------|-----------------------|------------|
| Stir-Verified-Request-Exception-Id | 107 | Yes | Yes | Yes | Yes | String |
| Stir-VS-Verstat | 108 | Yes | Yes | Yes | Yes | UTF8String |
| Stir-VS-Reason | 109 | Yes | Yes | Yes | Yes | String |
| Stir-TN-Used-For-AS-VS-Request | 110 | Yes | Yes | Yes | Yes | String |
| Stir-Div-Signed-Request | 111 | Yes (If generate-start = OK) | Yes | Yes | Yes | String |
| Stir-Div-Verified-Request | 112 | Yes | Yes | Yes | Yes | String |
| Stir-VS-Invite-State | 118 | Yes | Yes | Yes | Yes | String |
| Stir-VS-Attest | 119 | Yes | Yes | Yes | Yes | String |
| Stir-Div-VS-Reason | 120 | Yes | Yes | Yes | Yes | String |
| Stir-Div-VS-Verstat | 121 | Yes | Yes | Yes | Yes | String |
| Stir-AS-Bypass-Token | 154 | Yes (On 200 OK of invite) | Yes | Yes | Yes | String |

Stir-Signed-Request AVP (104)

This AVP indicates that the SBC has sent the signing request to the STI-AS and received a response. Values for this AVP mean:

- 1— The SBC has signed the INVITE.
- 0—The SBC has not signed the INVITE.
- 2—Signing is not applicable for the INVITE.

Stir-Signed-Request-Exception-Id AVP (105)

This AVP indicates that the SBC did not sign the request because of a failure request from the STI-AS. It includes the ATIS defined service or policy exception code. Potential values are in the next table.

| Exception ID | Exception Text | HTTP Status Code |
|--------------|---|------------------|
| SVC4000 | Missing request body. | 400 |
| SVC4001 | Missing mandatory parameter '%1' | 400 |
| SVC4002 | Requested response body type '%1'is not supported | 406 |

| Exception ID | Exception Text | HTTP Status Code |
|--------------|---|------------------|
| SVC4003 | Requested resource was not found | 404 |
| SVC4004 | Unsupported request body type, expected '%1' | 415 |
| SVC4005 | Invalid '%1' parameter value: %2 | 400 |
| SVC4006 | Failed to parse received message body: %1 | 400 |
| SVC4007 | Missing mandatory Content-Length header 411 | 411 |
| POL4050 | Method not allowed | 405 |
| POL5000 | Internal Server Error. Please try again later | 500 |

Stir-Verified-Request AVP (106)

This AVP indicates that the SBC has sent the verification request to the STI-VS and received a response. Values for this AVP mean:

- 1— The SBC has verified the request.
- 0—The SBC has not signed verified the request.
- 2—Request verification is not applicable for the call.

Stir-Verified-Request-Exception-Id AVP (107)

This AVP indicates that the SBC did not verify the request because of a failure request from the STI-VS. It includes the ATIS defined service or policy exception code. Potential values are in the next table.

| Exception ID | Exception Text | HTTP Status Code |
|--------------|--|------------------|
| SVC4000 | Missing request body | 400 |
| SVC4001 | Missing mandatory parameter '%1' | 400 |
| SVC4002 | Requested response body type '%1' is not supported | 406 |
| SVC4003 | Requested resource was not found | 404 |
| SVC4004 | Unsupported request body type, expected '%1' | 415 |
| SVC4005 | Invalid '%1' parameter value: %2 | 400 |
| SVC4006 | Failed to parse received message body: %1 | 400 |
| SVC4007 | Missing mandatory Content-Length header 411 | 411 |
| POL4050 | Method not allowed | 405 |
| POL5000 | Internal Server Error. Please try again later | 500 |

Stir-VS-Verstat (108)

The SBC populates the Stir-VS-Verstat attribute with the verstat values the SBC includes in the SIP INVITE following the Stir Shaken trigger with the values include TN-Validation-Passed, TN-Validation-Failed or No-TN-Validation in the cases below. DIV verification apply to 3GPP mode only.

The SBC includes a verstat in the INVITE and CDR:

- For SHAKEN verifications, when the SBC receives a verstat value parameter in the HTTP response from STI-VS server.
- For DIV verifications, when the SBC receives multiple verstat value parameters in a single HTTP response from the STI-VS server, and a specified final verstat value.
- For both DIV and SHAKEN verifications, when the SBC does not send a VS request to the STI-VS server, it adds the No-TN-Validation verstat to the egress INVITE. This may happen, for example, when the ingress INVITE doesn't have the Identity header.
- For both DIV and SHAKEN verifications, when the SBC does not receive any response from the STI-VS server, it adds the No-TN-Validation verstat to the egress INVITE. An example of this is a timeout.
- The SBC populates Stir-VS-Verstat with an empty string in the following cases:
 - For all authentication scenarios
 - If there is a timeout
 - If the system does not send a request to the STI-VS because there is an existing Identity header
 - For verification scenarios wherein the system receives a 4xx/5xx response from the STI-VS because the verstat and reason fields are not included in the json response
 - If the STI-VS response does not have a Verstat Value
 - For ATIS deployments, if the STI-VS response has empty reasoncode and reasontext fields

Enable the **account-config, cdr-output-inclusive** parameter to include the empty strings for the scenarios above. If disabled, the system does not include the AVP in the CDR file.

Note

Unlike RADIUS, the SBC does not limit custom verstat values conveyed over DIAMETER to 30 bytes.

Stir-VS-Reason (109)

From a high level, this AVP reflects what the SBC includes in the SIP egress INVITE following the Stir Shaken trigger. The SBC populates the Stir-VS-Reason attribute with received reason information in the cases below, based on operating mode.

- When operating in ATIS mode, if you have enabled the **reason-json-sip-translation** feature and the system receives the reasoncode and reasontext parameters in the HTTP response
- When operating in 3GPP mode, if you have enabled the **reason-json-sip-translation** feature and the system receives an error response, either 4xx or 5xx, from the STI-VS, the

SBC uses the HTTP error code and reason phrase (if available) to populate a reason header in the egress INVITE

The SBC populates Stir-VS-Reason with an empty string in the following cases:

- When the **reason-json-sip-translation** feature is disabled
- For all authentication scenarios, regardless of whether the feature is enabled or disabled
- If there is a timeout
- If the system does not send a request to the STI-VS because there is an existing Identity header
- For both DIV and SHAKEN verifications, when the feature is enabled and the system receives a 4xx or a 5xx verification response because the verstat and reason fields are not included in the json response

Note

This is only true for ATIS 3GPP deployments.

Enable the **account-config**, **cdr-output-inclusive** parameter to include the empty strings for the scenarios above. If it's disabled, the system does not include the AVP in the CDR file.

Stir-TN-Used-For-AS-VS-Request (110)

This AVP contains the TN number captured by the SBC. The TN to be used in the CDRs uses the following priority order:

1. From the Tel PAI (if present)
2. From the SIP PAI (if present, and Tel PAI is not present)
3. From the "From" header if both Tel and SIP PAI are not present.

Stir-Div-Signed-Request (111)

Upon sending the div signing request to the STI-AS and receiving a 200 OK successful response from STIR-AS, the SBC enumerates the results as follows:

- "1" for div signing the INVITE
- "0" for not div signing the INVITE
- "2" when div signing is not applicable on the INVITE

Stir-Div-Verified-Request (112)

Upon sending the div verify request to the STI-AS and receiving a 200 OK successful response from STIR-VS, the SBC enumerates the results as follows:

- "1" for div verification of the INVITE
- "0" for no div verification of the INVITE
- "2" when div verification is not applicable on the INVITE

Stir-VS-Invite-State (118)

The SBC populates this attribute only if the attribute has a valid value extracted from the signaling, such as Stir-VS-Invite-State: "Terminated".

If there is no value extracted for this attribute, the SBC leaves this field empty.

When collecting traffic for server configured as an STI-VS, the SBC populates this field with "Terminated" for rejecting the INVITE due to call rejection, and with "Continued" for when the SBC does not reject the INVITE. When collecting for an STI-AS, the SBC leaves this field empty.

Stir-VS-attest (119)

For 3GPP, the SBC implements the "Stir-Vs-attest" AVP to populate the attest claim value from the JSON body verification response returned by STI-VS. If there is no attest claim in the JSON body verification response, the system populates Stir-VS-Attest with a null value.

The system encloses the attest value string of the CDR attribute Stir-VS-Attest in double quotes. The ACME Diameter Attribute value is 119.

Valid values for this attribute include "A" OR "B" OR "C".

Note

The system populates this AVP only when you enable the MAN compliance option in the **sti-config**.

Stir-Div-VS-Reason (120)

The SBC implements the "Stir-Div-VS-Reason" AVP to populate the Div Verify result from the JSON body verification response returned by STI-VS.

The ACME Diameter Attribute value is 120. The system includes reasoncode and reasontext for Div Verify result, even if they are same or different in verify results.

Note

The system populates this AVP only when you enable the MAN compliance option in the **sti-config**.

Field population processes include:

- In 2xx case, Sti-div-VS-Reason is reasoncode + reason text from JSON body:
 - reasonCode (included in JSON body)
 - OR empty (if response code is not available in JSON)
 - reasonText (included in JSON body)
 - OR empty (if response text is not available in JSON)
- In 4xx/5xx case, Sti-div-VS-Reason is reasoncode + reason text from JSON body:
 - For reasonCode:

- * reasonCode (included in JSON body)
- * OR HTTP response code if reasonCode is not available in JSON body
- * OR empty (if HTTP response code not available)
- For reasonText:
 - * reasonText (included in JSON body)
 - * OR HTTP reason phrase if reasonText is not available in JSON body
 - * OR empty (if HTTP reason phrase is not available)

Stir-Div-VS-Verstat (121)

The SBC implements the “Stir-Div-VS-Verstat” AVP to populate the verstat value from the JSON body verification response returned by STI-VS. The system encloses the values for each verified DIV PASSporT separated with comma in double quotes. The ACME Diameter Attribute value is 121. Valid values include "TN-Validation-Passed, TN-Validation-Failed, and TN-Validation-Passed". The system includes each DIV verification result value even if the same code or verstat can be seen in different verification results.

Note

The system populates this AVP only when you enable the MAN compliance option in the **sti-config**.

Field population for the stir-div-vs-verstat AVP is closely related to population of the stir-vs-verstat AVP. Field population processes include:

- If both verstatValue and divResult verstat values are present:
 - Populate verstatValue for “stir-vs-verstat”.
 - Populate divResult verstatValue for “stir-div-vs-verstat”.
- If only verstatValue is present and no divResult, but ppt=div is there:
 - Populate verstatValue for “stir-vs-verstat”.
 - Populate verstatValue for “stir-div-vs-verstat”.
- If only verstatValue is present and no divResult, and ppt=div is not there, populate verstatValue for “stir-vs-verstat” only.

Stir-AS-Bypass-Token

This AVP contains the static bypass token added to the P-Identity-Bypass header in the egress INVITE for calls that bypassed the STI-AS process due to matching a configured bypass scenario.

The system encloses the bypass token value string in double quotes. The ACME Diameter Attribute value is 154. If there is no bypass token sent, the system does not include this AVP.

Acme-P-VSA-NNN (123-153)

These AVPs are vendor-specific UTF8 strings (D_AVP_ACME_VSA_200 to D_AVP_ACME_VSA_230), which use the AVP codes 123-153, populated by header

manipulation rules (HMR) with custom information from SIP message. They are included in the Acme-Packet-Specific-Extension-Rf (AVP Code 1) grouped AVP.

These AVPs use the following ABNF grammar.

```
< Acme-P-VSA-NNN>:: = < AVP Header: YYY >
[ AVP_ACME_VSA_NNN ]
```

where:

- *NNN* is the VSA number, from 200-230
- *YYY* is the corresponding AVP code, from 123-153

By default, the SBC does not populate these AVPs. You configure rules in the **sip-manipulation** element in the session-router path to enable this behavior.

ACR Event Records

The SBC supports ACR [Event] records, according to 3GPP TS 32.260. This is in addition to start, stop and interim records. These records reflect a preset type of SIP event. The ACRs are then sent to the CCF.

The SBC can create Event ACR messages on REGISTER, local-re-REGISTER and/or Short Message Service (SMS) message requests. A local re-register is when registration caching is enabled and the REGISTER from an currently registered endpoint occurs before half of the registration expiration time. In such cases the SBC sends a 200 OK to the re-registering endpoint and does not forward the re-REGISTER to the registrar. A message is an SMS event, on which the SBC collects critical RF information.

Event record creation is enabled with the **generate-event** parameter in the account config. This parameter can be set to register, local-register, message or any combination of all three values. The configured value(s) indicates the type of message that initiates an event ACR message sent to a CCF. Register only prompts the SBC to create Event ACRs at a REGISTER. Local-register prompts the SBC to create Event ACRs on a re-REGISTER that was replied to by the SBC. Message prompts the SBC to create EVENT ACRs sending information on SMS traffic within STOP records.

Event messages are created when the SBC receives a SIP Final Response 2xx or SIP Final Response (4xx or 5xx), that indicates an unsuccessful REGISTER.

Event ACR messages are also generated to indicate there has been an unsuccessful session attempt. Upon receiving a 4xx, 5xx or 6xx response to a SIP Invite, an Event message is created.

ACR Event Message Construction

An Event ACR is generated according to the tables that describe the AVPs present in the SBC's ACR message. Refer to the checked items in the Event column to see all included AVPs. Note that the Accounting-Record-Type AVP (480) is set to Event_Record (1).

Event-Type AVP

The Event-Type AVP (AVP code 823) is a Grouped AVP and contains information about the type of chargeable telecommunication service/event for which the accounting-request and/or credit control request message(s) is generated.

It is used in an AAR Event record. In this context, refer to the following:

| AVP | Number | Acme # | Definition |
|----------------|--------|--------|---|
| [SIP-Method] | 824 | 173 | Contains the name of the SIP Method. Values include REGISTER or MESSAGE (for SMS). These generate an accounting request to the CCF. |
| [Event] | 825 | 245 | Holds the content of the "Event:" header. This is not present in a REGISTER or re-REGISTER message |
| [Expires] | 888 | 246 | Holds the content of the "Expires" header. Upon a re-REGISTER this value is the time remaining until the endpoint's registration expires. |

The SBC generates additional information for SMS and distributes the information within the SIP-Method grouped AVP.

Expires Value

A refresher on the expires value: If the Contact: header does not contain an expires parameter, the SBC adds one with the value in the Expires: header in the 200 OK returned to the UA who sent the INVITE. If there is no Expires: header, the SBC adds expires parameter with a value of 3600 and an Expires header with a value of 3600 to the 200 OK.

As the SBC forwards the final 200 OK to the initiating endpoint, the Expires (888) AVP contains the largest expires value of all expires parameters in Contact: headers.

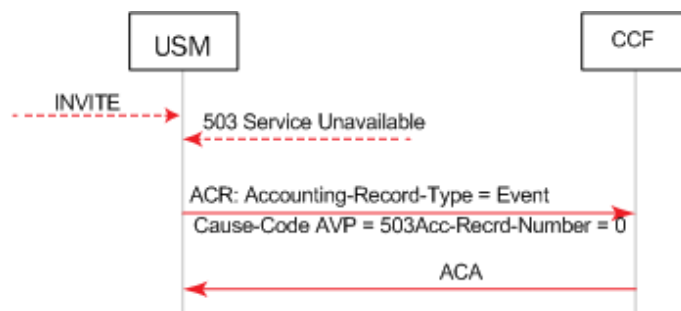
When registration caching is enabled and the SBC receives a reREGISTER from an endpoint before the halftime of the local registration has expired, the SBC inserts the maximum of all associates contacts' remaining time until expiry in the Expires AVP.

Event ACRs Generated for Unsuccessful Session Attempts

When any of the following responses are received in response to a SIP Invite, an Event ACR message is created to indicate there has been an unsuccessful session attempt:

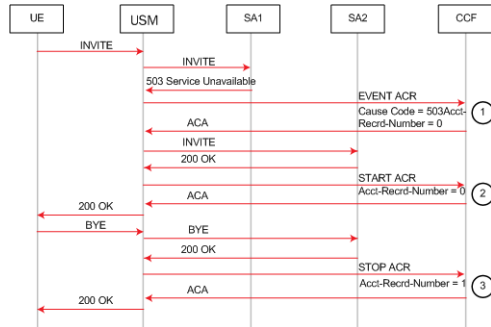
- 4xx Request Failure Code — this code is used when the SIP transaction is terminated due to an IMS node receiving a 4xx error response.
- 5xx Server Failure Code — this code is used when the SIP transaction is terminated due to an IMS node receiving a 5xx error response.
- 6xx Global Failure Code — this code is used when the SIP transaction is terminated due to an IMS node receiving a 6xx error response.

This example call flow shows how a 5xx server failure results in the sending of an Event ACR.



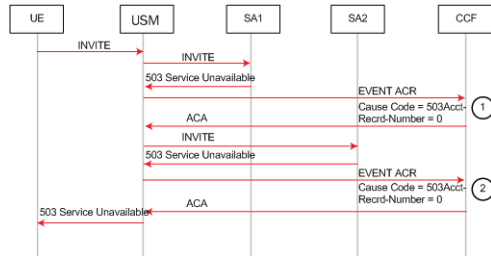
Call Flow Example Showing Event ACR Generation Due to 5xx Server Failure

The following call flow example shows two session agents. The first session agent replies with 503, which results in an Event ACR with cause code 503. The second session agent replies with 200OK that causes the sending of a Start ACR. The generate-start parameter is OK.



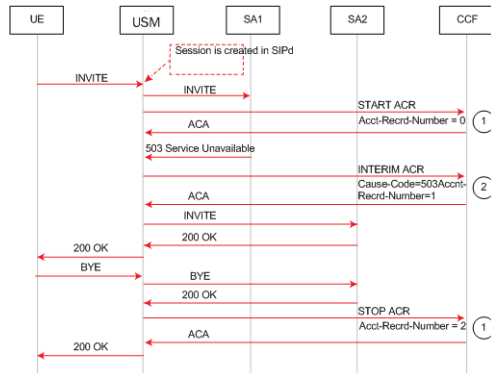
Call Flow Example Showing Sending of an Event and Start ACR

This example call flow illustrates the same call flow when generate-start are equal to Invite. In this case, Start, Interim and Stop ACRs are sent. The generate-start parameter is OK.



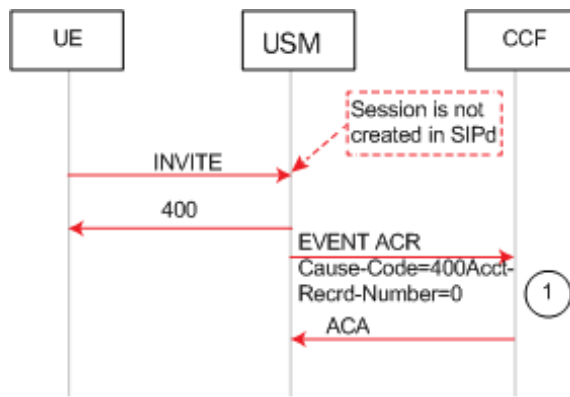
Call Flow Example Showing Generation of Event ACRs Due to 5xx Server Failures

This example call flow shows how a session is created and a Start ACR is sent but then a 5xx server failure occurs that results in sending an Interim ACR. The generate-start parameter is Invite.



Call Flow Example Showing Sending of Start and Interim ACRs

This example call flow shows how the SD rejects a call before a session is created. When the enhanced-cdr option is not set, an Event ACR is then sent.



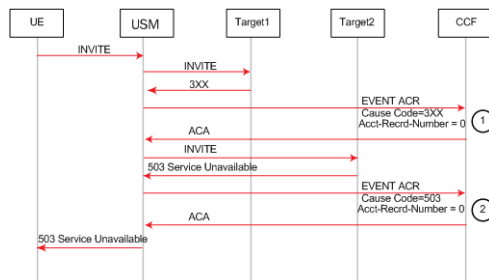
Call Flow Example Showing Unsuccessful Session Creation and Sending of a Event ACR

Event ACRs Generated for Receipt of SIP Messages

An Event ACR is issued when the following SIP messages are received:

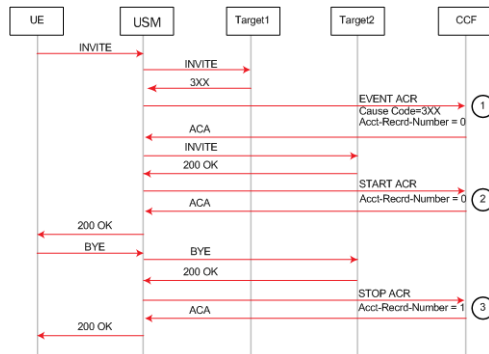
- 200 OK message for a SIP Register from the IMS core.
- SIP Final/Redirection Response 3xx
- SIP Final Response (4xx, 5xx or 6xx) — this indicates an unsuccessful session-unrelated procedure.
- SIP Final Response (4xx, 5xx or 6xx) — this indicates an unsuccessful SIP session set-up.

This example call flow shows an Event ACR that is created due to the reception of a 3xx SIP message. In this example the call fails after being redirected.



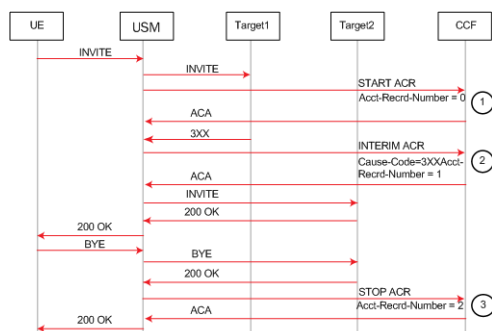
Event ACR Created Upon Reception of a 3xx Message Followed by Call Failure After Redirect

This example call flow shows an Event ACR that is created due to the reception of a 3xx SIP message. Then it shows a Start ACR upon reception of a 200OK. In this example the call succeeds after being redirected. The generate-start is OK.



Event ACR Created Upon Reception of a 3xx Message Followed by Call Success After Redirect

This example call flow shows a Start ACR being sent when an Invite is received and an Interim ACR that is created due to the reception of a 3xx SIP message. In this example the call succeeds after being redirected. The generate-start is Invite.



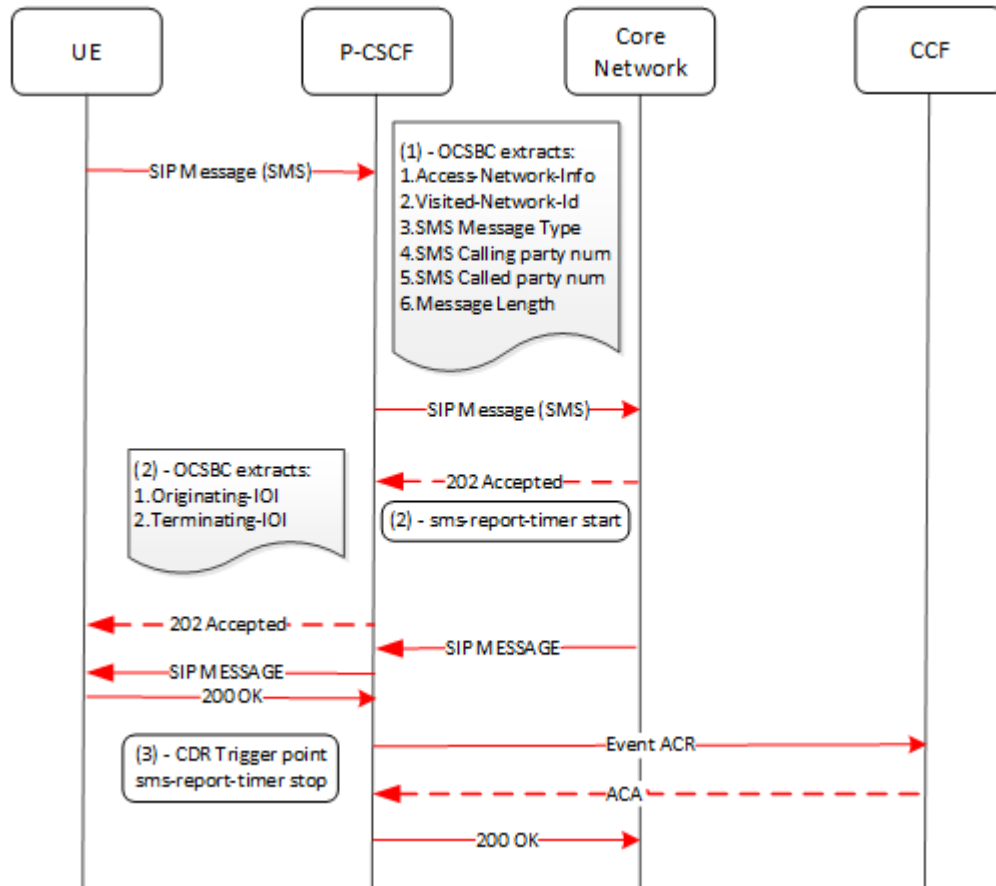
Event ACRs for SMS

The SBC generates call data event records with information for SMS that differs somewhat from that of calls. You configure the SBC to generate ACRs for SMS using the same procedures you use for calls, which then treats the SMS messages as events, similar to call data during registrations.

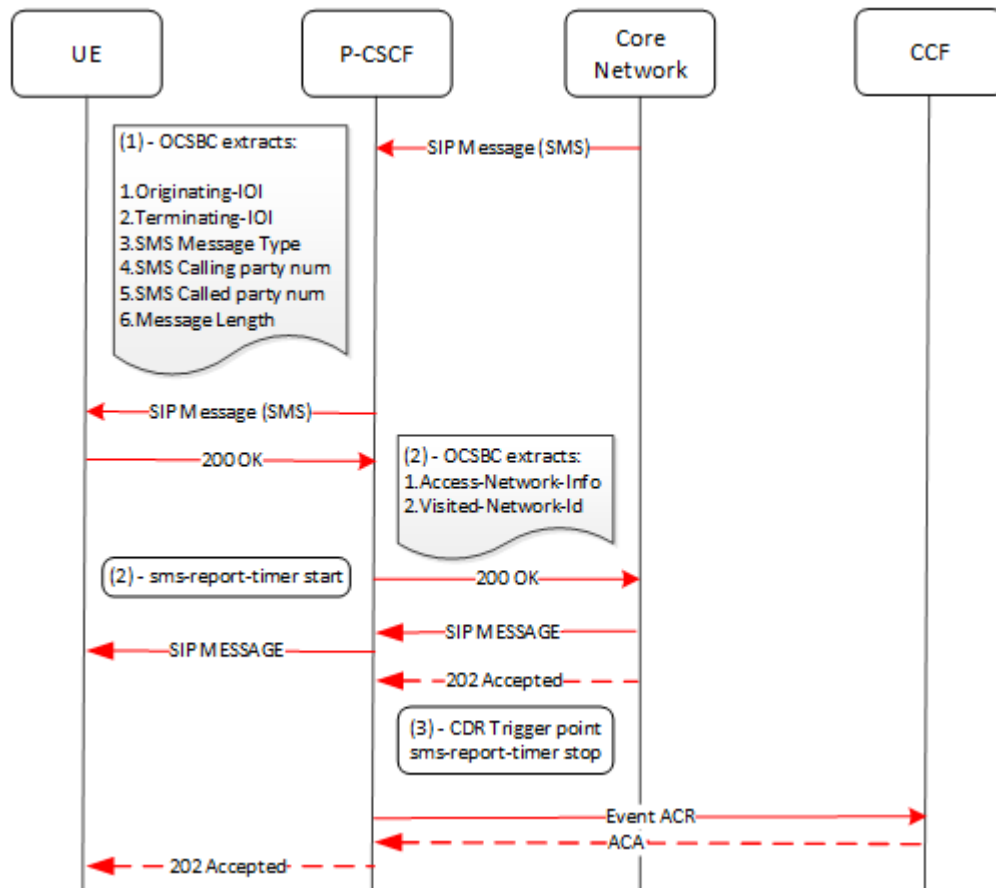
End stations send SMS messages using a SIP MESSAGE (content-type=application/vnd.3gpp.sms) with a GSM-SMS body. The SBC complies with 3GPP specifications 23.040 (Technical realization of the Short Message Service (SMS) and 24.011 Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface) with respect to message format. The SBC parses message information for AVP data during message processing and ultimately provides ACRs to the CCF using the Rf interface.

SMS processing that includes ACR generation includes a separate timer, the **sms-report-timeout** within the **sip-config**, for SMS accounting. If the SBC does not receive a delivery report /submit report within this timer's value, it discards all accounting for that message. The diagrams below present the start and stop locations of this timer within the context of overall ACR generation.

The call flow diagram below illustrates the SBC, as P-CSCF, gathering data for the SMS report during a message originating scenario. The SBC ultimately issues the ACR after the message delivery is complete.



The call flow diagram below illustrates the SBC, as P-CSCF, gathering data for the SMS report during a message terminating scenario.



VoLTE Call and SMS AVPs for Diameter

The SBC issues custom AVPs that capture VoLTE and SMS information within ACRs. These are in addition to the common AVPs for all traffic. The process for VoLTE calls is the same as all other calls. For SMS, the process is different given that the message is an event, and the data recorded on the message equates to what is recorded for a call session.

The table below identifies AVPs specific to VoLTE and SMS traffic.

| AVP | ACME Diameter Attribute | Start | Interim | Stop | Event = MESSAGE | AVP Type |
|--------------------------|-------------------------|-------|---------|------|-----------------|------------|
| Pgw-IP | 95 | Y | Y | Y | N | UTF8String |
| Sgw-IP | 96 | Y | Y | Y | N | UTF8String |
| IMEI | 97 | Y | Y | Y | Y | UTF8String |
| IMSI | 98 | Y | Y | Y | Y | UTF8String |
| History-Info | 99 | Y | Y | Y | N | UTF8String |
| Sms-Msg-Type | 100 | N | N | N | Y | UTF8String |
| Sms-called party-Number | 101 | N | N | N | Y | UTF8String |
| Sms-calling party-Number | 102 | N | N | N | Y | UTF8String |

| AVP | ACME Diameter Attribute | Start | Interim | Stop | Event = MESSAGE | AVP Type |
|----------------|-------------------------|-------|---------|------|-----------------|------------|
| Sms-Msg-Length | 103 | N | N | N | Y | Unsigned32 |

Diameter AVPs for VoLTE Calls

The SBC sends an ACR to the PCRF for call accounting with the following VoLTE-specific AVPs. The table shows all mandatory and optional AVP's. If there is data, the SBC includes Optional AVPs. If not the SBC does not include them.

| AVP Name | AVP Code | Is grouped? Group hierarchy | Type |
|--------------------------------|----------|--|--------|
| Access-Network-Information | 1263 | Yes [ACR] [Service-Information] [IMS Information] [Access-Network-Information] | String |
| IMS-Visited-Network-Identifier | 2713 | Yes [ACR] [Service-Information] [IMS Information] [IMS-Visited-Network-Identifier] | String |
| Originating-IOI | 839 | Yes [ACR] [Service-Information] [IMS Information] [Inter-Operator-Identifier] [Originating-IOI] | String |
| Terminating-IOI | 840 | Yes [ACR] [Service-Information] [IMS Information] [Inter-Operator-Identifier] [Terminating-IOI] | String |

In addition, the SBC sends the following fields as custom AVP's in the ACR.

| AVP Name | AVP Code | Type |
|----------------|----------|------------|
| IMSI | 98 | UTF8String |
| IMEI | 97 | UTF8String |
| History-Info | 99 | UTF8String |
| PGW-IP Address | 95 | UTF8String |
| SGW-IP Address | 96 | UTF8String |

Diameter AVPs for SMS Messages

The SBC sends an ACR to the PCRF for MESSAGE accounting with the following SMS-specific AVPs. The table shows all mandatory and optional AVP's. If there is data, the SBC includes Optional AVPs. If not the SBC does not include them.

| AVP Name | AVP Code | Is grouped? Group hierarchy | Type |
|--------------------------------|----------|--|--------------|
| Session ID | 263 | No | String |
| Origin Host | 264 | No | String |
| Origin Realm | 296 | No | String |
| Destination Realm | 283 | No | String |
| Origin State ID | 278 | No | Unsigned Int |
| Accounting Record Type | 480 | No | Enum |
| Accounting-Record-Number | 485 | No | Unsigned Int |
| Accounting App ID | 259 | No | Unsigned Int |
| Event Timestamp | 55 | No | Time |
| Service Context ID | 461 | No | String |
| Subscription-Id-Type | 450 | Yes [ACR] [Service-Information] [Subscription ID] [Subscription-Id-Type] | Enum |
| SIP-Method | 824 | Yes [ACR] [Service-Information] [IMS Information] [Event-Type] [SIP-Method] | String |
| Role-Of-Node | 829 | Yes [ACR] [Service-Information] [IMS Information] [Role-Of-Node] | Enum |
| Node Functionality | 862 | Yes [ACR] [Service-Information] [IMS Information] [Node Functionality] | Enum |
| SIP-Request-Timestamp | 834 | Yes [ACR] [Service-Information] [IMS Information] [Timestamp] [SIP-Request-Timestamp] | Time |
| SIP-Response-Timestamp | 835 | Yes [ACR] [Service-Information] [IMS Information] [Timestamp] [SIP-Response-Timestamp] | Time |
| SIP-Request-Timestamp-Fraction | 2301 | Yes [ACR] [Service-Information] [IMS Information] [Timestamp] [SIP-Request-Timestamp-Fraction] | Unsigned Int |

| AVP Name | AVP Code | Is grouped? Group hierarchy | Type |
|---------------------------------|----------|---|--------------|
| SIP-Response-Timestamp-Fraction | 2302 | Yes [ACR] [Service-Information] [IMS Information] [Timestamp] [SIP-Response-Timestamp-Fraction] | Unsigned Int |
| Access-Network-Information | 1263 | Yes [ACR] [Service-Information] [IMS Information] [Access-Network-Information] | String |
| IMS-Visited-Network-Identifier | 2713 | Yes [ACR] [Service-Information] [IMS Information] [IMS-Visited-Network-Identifier] | String |
| Originating-IOI | 839 | Yes [ACR] [Service-Information] [IMS Information] [Inter-Operator-Identifier] [Originating-IOI] | String |
| Terminating-IOI | 840 | Yes [ACR] [Service-Information] [IMS Information] [Inter-Operator-Identifier] [Terminating-IOI] | String |

In addition, the SBC sends the following fields as custom AVP's in the ACR.

| AVP Name | AVP Code | Type |
|--------------------------|----------|------------|
| IMSI | 98 | UTF8String |
| IMEI | 97 | UTF8String |
| SMS MSG Type | 100 | UTF8String |
| SMS Calling party number | 102 | UTF8String |
| SMS Called party number | 101 | UTF8String |
| Message Length | 103 | Unsigned32 |

Distinct VoLTE Processes

For VoLTE calls, the process for generating CDRs is the largely the same as for other calls. As described, there are additional data points included for these call types.

In addition, the list below presents additional processes reserved for VoLTE data management with which you should be familiar:

- When there is an SRVCC event, the SBC creates a separate set of CDRs for the handover session. The SBC correlates the original and handover session using the "Generic-ID"

field, which points to the Call-ID of the initial session. In addition, the SBC populates the Generic-ID field within the Initial Session CDRs (STOP), with the HO session Call-ID.

- The SBC copies the Call id of the second INVITE (Handover INVITE) into the Generic Id into the CDR for the first INVITE (initial call) for both MO and MT call
 - For mobile originating call—When the SBC receives the 200 Ok for the BYE from UE, it inserts the Call id of second INVITE, which is generated from the MSC-S as Generic Id, into the CDR of First MO Invite (Before the handover call).
 - For mobile terminating call—When the SBC receives the 200 Ok for the BYE from UE, it inserts the Call id of the second INVITE, which is generated from the MSC-S as Generic Id, into the CDR of the first MT INVITE (before the handover call).
 - If there is a negative case, such as a BYE timeout, the SBC writes the Call id of second INVITE, which is generated from the MSC-S as the Generic Id, into the CDR of the first INVITE (before the handover call) when that corresponding call gets terminated.

Note

The SBC performs these same processes for both RADIUS accounting when generating CDRs and Diameter accounting when generating ACRs.

Configurations to Specify VoLTE and SMS Data

You can configure the SBC to use specific data in call data records provided over RADIUS, Diameter and within local CSVs. This ensures that the specified fields

There are two configurations available for specifying VoLTE and SMS data:

- Subscribe to IP-CAN-CHANGE events
- Specify Inter-Operator Identifier (IOI)

Subscribe to IP-CAN-CHANGE Events

You can configure a subscription to the IP-CAN-CHANGE event using the Rx interface during the AAA/RAR exchange. To do this, you configure the **ip-can-change** value to the **specific-action-subscription** of the applicable **ext-policy-config**. This causes the SBC to apply the value in the AN-GW-Address AVP from the PCRF as the S-GW IP address.

Note

If the SBC receives more than one AN-GW-Address AVP from the PCRF, it applies the value in the AN-GW-Address AVP from the PCRF. It also uses the S-GW IP address the first AVP as the S-GW IP address.

Option to Specify IOI

You configure the **realm-as-ioi** option in the applicable **account-config** to send the realm name as the IOI in diameter ACRs. If this option is not set, the SBC uses the IOI from the charging vector.

Configure this option using the syntax below.

```
ORACLE(account-config)# options +realm-as-ioi
```

If you type options and then the option value without the plus sign, you overwrite any previously configured options. To add a new option to an options list, pre-pend the new option with a plus sign as shown in the previous example.

Configuring the ip-can-change Subscription

You use the steps below to Subscribe to IP-CAN-CHANGE events at the PCRF and apply the value in the AN-GW-Address AVP from the PCRF as the S-GW IP address.

To obtain the S-GW IP address for mobile originating and terminating scenarios and provide that data in CDRs for RADIUS, diameter and local CSVs:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ORACLE# configure terminal
```

2. Type **media-manager** and press Enter.

```
ORACLE(configure)# media-manager
```

3. Type **ext-policy-config** and press Enter.

```
ORACLE(session-router)# ext-policy-config
ORACLE(ext-policy-config)#
```

4. Type **specific-action-subscription, ip-can-change** and press Enter.

```
ORACLE(session-router)# specific-action-subscription ip-can-change
ORACLE(ext-policy-config)#
```

The **specific-action-subscription** accepts multiple values. When configuring 2 or more specific actions, enclose them in quotation marks, with the values separated by spaces.

5. Save your work.

Event Local CSV File

This feature also creates an Event CSV file when the **generate-event** parameter is enabled. The following table describes the inclusive CSV element order:

| CSV Placement | Attribute Name | AVP Number |
|---------------|---------------------|------------|
| 1 | Record Type | 480 |
| 2 | NAS IP Address | |
| 3 | NAS Port | 16 |
| 4 | Calling Station ID | |
| 5 | Called Station ID | |
| 6 | Diameter Session ID | 263 |
| 7 | SIP Method | 824 |
| 8 | Event Time | 55 |

| CSV Placement | Attribute Name | AVP Number |
|---------------|---------------------|------------|
| 9 | User Name | 1 |
| 10 | Node Function | 862 |
| 11 | Application ID | 259 |
| 12 | Role Node | 829 |
| 13 | Event | 825 |
| 14 | Expires | 888 |
| 15 | Associated URI | 856 |
| 16 | Cause Code | 861 |
| 17 | CDR Sequence Number | |
| 18 | Origin Realm | 296 |
| 19 | Origin Host | 264 |
| 20 | Destination Realm | 283 |
| 21 | Destination Host | |

The SBC generates a different CSV for message events, triggered by SMS traffic. This layout is presented in Appendix B.

Diameter Heartbeat for Rf

Device-Watchdog-Request (DWR) and Device-Watchdog-Answer (DWA) messages are used to detect transport failures at the application layer between the SBC communicating with a policy server via Diameter. The request/answer message pair forms a heartbeat mechanism that can alert the requesting side if the answering side is not reachable.

The SBC always responds to a DWR by replying with a DWA message. In addition, the SBC can be configured to initiate DWR messages toward a policy server or other Diameter-based network device.

You configure the **watchdog ka timer** to indicate the period between the DWRs the SBC sends to a Diameter Agent, an Rf based server in this case.

If the SBC fails to receive a DWA response from the Server within 1/4 of the configured **watchdog ka timer** parameter, then the connection towards that Server is considered failed and torn down. The SBC attempts to recreate the TCP connection, followed by the recreating the Diameter connection by issuing a CEA toward the policy server.

When other Rf traffic to/from the accounting server is present, DWRs are suspended. The other traffic indicates that the server is up. The SBC upon detection of DWR failure can send accounting data towards another configured accounting server by failover/strategy mechanisms.

Using FQDNs to Access CCFs over Diameter

You can configure the SBC with a primary and, if desired, a secondary FQDN to access CCF servers over Diameter. You do this by configuring the diameter **account-server** with an FQDN. The SBC uses DNS to resolve the FQDN into an IP list and, if provided, route the traffic based on DNS-provided priority and weight. The SBC supports resolution of CCF FQDNs from SRV, and A records.

Upon configuration, the SBC can perform a DNS resolution procedure to resolve a configured FQDN to one or more IP addresses. When the SBC receives the DNS response, it extracts the IP address(es). Each IP address corresponds to one CCF. For each IP address, the SBC next

establishes a TCP connection and performs a CER and CEA exchange, which makes that CCF eligible to exchange Rf traffic. Having established a connection with these servers, the SBC performs the same Diameter procedures and processes it performs when configured with an IP address.

When you configure an **account-server**, you can specify the server **hostname** with an FQDN instead of an IP address. In addition, you specify the realm to use for DNS queries. Upon activation, the SBC uses DNS to:

- Identify one or more IP addresses that correspond to your FQDN.
- Utilize the port provided in the response.

Note

If there is no port or port 0 in the response, the SBC contacts the CCF via the port you configure in your **account-server**.

- Utilize DNS-provided TTL to start the timer used to determine when to re-send an SRV query.
- Utilize DNS-provided priority and weight to determine the order with which to send ACR traffic.

Note

If there is no priority or weight in the DNS response, the SBC uses round robin to define the order it uses to contact CCFs.

- Provide dynamic updates to CCF servers, and perform CCF addition or deletion.
- Achieve runtime scaling of CCF servers without additional configuration.

Ensuing SBC behavior is dependent on the DNS response:

- Single IP address received—Priority is irrelevant and the SBC simply uses the address provided as the only CCF target.
- Multiple IP addresses received—The receipt of multiple addresses makes multiple CCFs available for operations without configuration changes. The SBC creates a table of CCF resolutions, including DNS priority and weightage, and creates a target list. It then creates TCP connections, performs CER/CEA exchanges with all IP addresses, and routes ACRs to CCFs using the established order.
- Error/No Response—If the SBC receives an error response or no response to the SRV-query, it starts an internal DNS retry timer before it attempts to contact the servers. Also, if it finds the primary DNS Server is down, the SBC retries using your configured secondary DNS Server.
- TTL below 30 secs—If the SBC receives a TTL that is less than 30 secs for any IP address, it uses 30 seconds as the TTL. This ensures that the system does not become overloaded by an incorrect configuration.
- Zero weightage—The SBC complies with RFC 2873 by ignoring any resolutions that include a weight of zero.

If the SBC receives 3002, 3004 or 4002 error response codes from the CCF, it retries sending the ACR.

The SBC provides traffic statistics on all the IP addresses resolved against an FQDN for both the primary and secondary pools.

Traps and Alarms

For error responses received against accounting or non-accounting requests from the CCF, including CER and DWR as well as ACR issues, the SBC issues SNMP traps and raises local alarms. Examples of these errors include determining the ACR/AVP format or content is malformed or unsupported. Examples of the applicable alarms are presented below.

| ID | Task | Severity | First Occurred | Last Occurred |
|--------|--|----------|---------------------|---------------------|
| 327708 | 3029 | 5 | 2021-11-16 10:31:18 | 2021-11-16 10:31:18 |
| Count | Description | | | |
| 1 | Diameter Accounting Server lost connection!!! | | | |
| 327709 | 3029 | 5 | 2021-11-16 10:33:24 | 2021-11-16 10:33:24 |
| Count | Description | | | |
| 1 | Diameter Accounting Server Returned Error Result Code 10.196.2.7:3869-3002 | | | |

The applicable Traps, within `ap-diameter.mib`, include `apDiameterSrvrErrorResultTrap` (1.3.6.1.4.1.9148.3.13.1.2.2.0.5) and `apDiameterSrvrSuccessResultTrap` (1.3.6.1.4.1.9148.3.13.1.2.2.0.6).

Configuration that Applies to FQDN CCFs

Key configuration on the `account-config` includes:

- The **dns-realm** parameter specifies the realm you use for resolving CCF FQDNs. This realm needs a network-interface with appropriate DNS configuration.
- Enabling the use of affinity for CCF sessions
The **maintain-ccf-affinity** parameter enables the SBC to establish CCF affinity for ACR sessions, simplifying the reconciliation of ACRs by keeping associated records on the same CCF. Affinity applies to CCFs that you configure with either FQDN and IP address. For FQDN, the SBC still uses the mechanisms described above to determine the first access to a CCF. For determining traffic targets after the first access, the SBC prioritizes affinity over pool membership and DNS priority/weightage.
To support affinity, the SBC keeps a mapping between CCF and ACR sessions so that it sends all new ACRs for an existing session to the same CCF. The SBC breaks affinity when a selected CCF goes down or is removed from the DNS response.
- The **next-priority-selection-interval** parameter, which specifies in minutes how long the system waits after experiencing an overload scenario before it refers to the next priority.
- The **send-disconnect-peer-msg** parameter, which enables the system to send a DPR to the CCF before disconnecting from the CCF.
- The **strategy** parameter specifies how to select from a pool's CCF servers to distribute CCF traffic. This configuration applies to FQDN CCFs when the DNS response does not include priority/weight.

Key configuration on the `account-sever` includes:

- The **hostname** parameter specifies the FQDN you use to fetch DNS resolution of your pool's CCF servers.
- The **fqdn-pool-type** parameter specifies whether the pool established by the DNS resolution is the primary or secondary pool of CCF servers.

Handling Multiple CCFs

When a DNS response includes multiple CCFs as potential targets, the SBC uses a selection procedure based on your configuration, the DNS response, and current traffic conditions.

When configuring for FQDN-resolved CCF access order, you begin with determining whether you want to establish a primary and a secondary pool of servers and, if so, which FQDN is primary. This configuration establishes the first criteria for determining access order. The process of checking all potential resolutions begins with checking the resolutions from the primary **account-server**. If the system does not reach a CCF after that, it performs the same process using the secondary **account-server**.

When selecting from a pool, the SBC first refers to DNS server priority to determine the access order. The lower the priority number, the earlier the CCF appears in the list. Having established an order based on priority, the SBC then refines that order using IP address. Specifically, the SBC sorts servers that have the same priority based on the returned IP addressing. In this case, the system selects the lowest IP address in the resolution and attempts to reach that one first. If there is no response, the system tries the next lowest IP address until it has attempted every resolution.

Having identified server priority from DNS, the SBC proceeds with forwarding call-generated ACR signaling sequences with the group of highest priority servers only. The SBC uses weight to establish a round-robin method of selecting CCFs for call-generated ACR sequences from the highest priority list until it has attempted to reach all CCFs in that list. If there are no highest priority CCFs available, the SBC starts using CCFs from the next highest priority list.

The system only moves to the next priority group when all CCFs in the first priority group are down or have reached their acr buffer thresholds.

The SBC also uses weight to determine how many calls it can send to a given CCF within a single call session cycle. A call session cycle is defined as reaching the number of calls equal to the sum of all CCF weights. The system uses weight to determine the number of calls it can send to a CCF during that call cycle. The system selects each CCF of the same priority in round robin, and removes each CCF from the cycle when they reach their weight. The cycle ends when each CCF has handled its maximum. Regardless of how many calls may still be active, the system restarts the cycle once it has sent the lowest weight CCF its maximum number of calls.

If, at any point in the call session cycle, the system finds all CCFs busy, based on buffer usage, it begins to use the next highest priority group. This is also true for the secondary pool. When triggered by buffer usage by the entire primary pool, the system begins to use the highest priority group in the secondary pool. In addition, the system moves on to the secondary pool if it finds all CCFs in the primary unavailable.

Buffer Overflow Scenario

If the SBC determines that configured buffer limits are exhausted, it starts routing traffic to lower priority or secondary pool CCFs to avoid buffer overflow.

Three configurations under **account-config** specify this behavior:

- **acr-buffer-upper-threshold**— Indicates the percentage of buffer usage beyond which the SBC routes new call sessions to alternate CCFs.
- **next-priority-selection-interval**— When buffer usage has exceeded the upper threshold, the system starts this timer. When the timer expires, the system checks buffer usage again. If buffer usage still exceeds the threshold, the system changes its selection criteria

to include the next lower priority CCFs or, if they are all in use, CCFs from the secondary-pool.

- **acr-buffer-lower-threshold**— Indicates the percentage of buffer usage that triggers the SBC to start using that CCF again.

When buffer usage exceeds your **acr-buffer-upper-threshold** value, the SBC:

1. Sends an SNMP trap and raises an alarm with information about the current usage of the ACRq along with the configured lower and upper threshold. The applicable alarm, named DIAM ACCT BUFFER THRESHOLD EXCEED, appears as follows with the ID a.

```
1 alarms to show
ID          Task          Severity      First Occurred      Last Occurred
327738      3029             6             2021-11-16 10:31:18 2021-11-16 10:31:18
Count      Description
1          Buffer Usage (90%) hit Upper Threshold Limit for Diameter
Accounting buffer!!!
```

The applicable traps, within ap-diameter.mib, include apAcctMsgQueueUpperThresholdTrap (1.3.6.1.4.1.9148.3.13.1.2.2.0.7) and apAcctMsgQueueUpperThresholdClearTrap (1.3.6.1.4.1.9148.3.13.1.2.2.0.7).

2. Starts routing calls to the CCFs from the next lower priority and continues routing existing sessions to the CCFs already assigned to them.
3. If the next lower priority CCFs are busy, the SBC selects CCFs with the priority below that for new calls.
4. If all priority CCFs are busy, the SBC selects CCFs from the secondary pool for all new sessions.
5. Continues using lower priority, and any secondary pool CCFs until buffer usage becomes equal or less than your **acr-buffer-lower-threshold** value.
6. Re-starts the **next-priority-selection-interval** timer each time buffer usage does not recover. On the second cycle through the **next-priority-selection-interval**, the SBC checks whether any of the highest priority CCFs are no longer servicing any sessions. If so, the system considers that (or those) CCFs available for service again, and places them at the top of the new session target list.

When buffer usage recovers, the system completes all sessions on existing CCFs and restarts assigning new sessions to the highest priority CCFs available in the primary pool.

Note

On expiry of usageCheck timer, SBC restarts the timer and then, instead of directly going for further lower priority, the system checks if any of the higher priority CCFs are available.

Furthermore, the SBC checks if outstanding ACRs of all the CCFs of any priority is zero and CCFs are eligible then SBC shall consider it as ready to serve.

The table below presents an example of a CCF distribution list.

| DnsResult(Primary) | | | | | DnsResult(Secondary) | | | | |
|--------------------|-------------|------|----------|-----------|----------------------|-------------|------|----------|-----------|
| CCF Number | hostname | port | Priority | Weightage | CCF Number | hostname | port | Priority | Weightage |
| CCF1 | 10.196.0.13 | 3868 | 1 | 2 | CCF11 | 10.196.0.43 | 3868 | 1 | 2 |
| CCF2 | 10.196.0.14 | 3868 | 1 | 3 | CCF12 | 10.196.0.44 | 3868 | 1 | 3 |
| CCF3 | 10.196.0.15 | 3868 | 1 | 6 | CCF13 | 10.196.0.45 | 3868 | 1 | 6 |
| CCF4 | 10.196.0.16 | 3868 | 1 | 9 | CCF14 | 10.196.0.46 | 3868 | 1 | 9 |
| CCF5 | 10.196.0.21 | 3868 | 2 | 30 | CCF15 | 10.196.0.51 | 3868 | 2 | 30 |
| CCF6 | 10.196.0.22 | 3868 | 2 | 60 | CCF16 | 10.196.0.52 | 3868 | 2 | 60 |
| CCF7 | 10.196.0.31 | 3868 | 3 | 6 | CCF17 | 10.196.0.61 | 3868 | 3 | 6 |
| CCF8 | 10.196.0.32 | 3868 | 3 | 9 | CCF18 | 10.196.0.62 | 3868 | 3 | 9 |
| CCF9 | 10.196.0.33 | 3868 | 3 | 12 | CCF19 | 10.196.0.63 | 3868 | 3 | 12 |

Having received resolution, the SBC routes the traffic in the CCFs primary pool with priority 1, which includes CCF1 to CCF4. If the SBC finds buffer usage has reached the upper threshold value, it starts selecting CCF5 and CCF6 for new sessions, while routing the existing sessions to CCF1 through CCF4.

Having cycled through CCF1 to CCF6, and after the usageCheck timer has expired, the SBC checks whether buffer usage has recovered:

- If CCF1 to CCF4 have no outstanding request, the system routes ensuing sessions to CCF1 to CCF4.
- If CCF1 to CCF4 still have outstanding requests, the system starts routing the traffic to CCF7 to CCF9.

If CCF1 to CCF9 are in use, and the timer has expired, the SBC attempts to use CCF1 to CCF4 again. If they are still busy, the system tries CCF5 and CCF6. If they are also busy, the SBC starts selecting from the secondary pool, specifically CCF11 to CCF14.

The SBC continues to monitor buffer usage, returning to the highest priority CCFs in the primary pool when it detects that usage has recovered. At this point, the SBC stops the usageCheck timer, sends an SNMP clear trap and clears the alarm.

Target List Updation

The SBC updates its CCF target list on a per-FQDN basis while minimizing DPRs and maintaining affinity when there is a:

- DNS cache refresh
- ACLI configuration changes

DNS refresh changes include:

- IP Address Added—The system assigns the new address to a specific pool, connects with CCF, and routes new call sessions to newly added IP also based on its priority and weight.

- **IP Address Removed**—The system stops sending new ACRs to the address and waits for ACAs to return until its **acr-retry-interval** expires. When it receives all the ACAs or the timer expires, the system sends DPR and disconnects the TCP connection. The system also ends any affinity processing for this CCF.
- **Port changes to an IP Address**—The system responds to this as if the applicable IP address was removed.
- **Priority/Weight change**—The system routes new call sessions based on modified priority without any impact on live sessions.

Configuration change impacts include:

- **Disabled `account-config`** —The system stops making DNS queries, and tears down all connections by sending a DPR (without checking the status of any outstanding ACRs).
- **Primary Pool Added**—This equates to enabling the **`account-config`**.
- **Secondary Pool Added**—You can only add a secondary pool if you have already configured the primary. In this scenario, the system makes the DNS query and sets up the secondary pool list
- **Secondary Pool deleted**—The system responds to this as if you have removed all secondary pool IP addresses.

Note

You cannot remove a primary pool.

- **FQDN changed**—The system makes a DNS query and gets the DNS result. Based on DNS result, the system compares the existing and the new lists and performs the same procedures as when an IP address is added or removed.
- **Pool Type changed**—The system treats this scenario as if there was an FQDN change.
- **Other than FQDN changed**—The system maintains CCF affinity and does not issue any DPRs.

DPR/DPA Support

The SBC supports transmission and reception of DPRs, which confirm disconnects and disables retries. DPR/DPA processing includes:

- **The CCF sends the DPR**—The system replies with a DPA. It then removes any affinity mapping and routes any outstanding ACRs to alternate CCFs. If there is an FQDN refresh for this CCF after a TTL, the system reconnects to it.
- **The SBC sends the DPR**—The system waits for all outstanding ACAs, based on the maximum retries. Assuming you have enabled the **`send-disconnect-peer-msg`** parameter, the system then removes any affinity mapping and routes outstanding ACRs to the next CCF.

Note

The system also performs this procedure when you have configured the **`account-server`** with an IP address.

If you have configure the **`account-server`** with an IP address, the system responds to DPRs from the CCF by:

- Responding with a DPA
- Disconnecting the TCP connection with CCF
- Removing any affinity mapping
- Starting the **restart-delay** per the **account-server** configuration

At this point, you can remove the CCF from the ACLI., otherwise the system tries to reconnect to that CCF when the timer expires.

Accounting Traffic and Server Statistics

Use the **show accounting** command to monitor the status and observe traffic statistics of your account servers.

The syntax for the **show accounting** command follows.

```
show accounting [<IPPort> | All] | [connections] | [ servers ]
```

Command arguments refine the output to address more specific detail, including:

- **<ipaddress:port>**—Reports on the server you specify with IP address and port. Specify these servers using the syntax **<ipaddress:port>**.
- **all**—Reports on all accounting servers
- **connections**—Shows connections for all accounting servers
- **servers**—Reports on servers when resolved using DNS and connection details for all Accounting Servers per pool

The output below presents the standard traffic data generated when using no argument, the **ipport** argument and the **all** argument.

```
SBC01# show accounting
07:23:40-198
Accounting Status:
-- Period --          ----- Lifetime -----
Active  High  Total  Total  PerMax  High
Request Queue      0      0      0      0      0      0
Client Trans       0      0      0      0      0      0
Server Trans       0      0      0      0      0      0
Sockets            2      2      0      2      2      2
Connections        0      0      0      0      0      0

Total Accounting Server Stats:
-----Lifetime -----
Recent  Total  PerMax
Msgs Queued      0      0      0
Msgs Discarded   0      0      0
Wait Queue Asks  0      0      0
Wait Queue Pops  0      0      0
Msgs Reclaimed   0      0      0
Msgs Sent        0      0      0
Msg Send Failed  0      0      0
Msgs ReSent      0      0      0
Msgs Rcvd        0      0      0
Msgs Processed   0      0      0
```

| | | | |
|--------------------|----|-------|---|
| Conn Timeouts | 13 | 20880 | 7 |
| Bad State Drops | 0 | 0 | 0 |
| Bad Type Drops | 0 | 0 | 0 |
| Bad Id Drops | 0 | 0 | 0 |
| Auth Failed Drops | 0 | 0 | 0 |
| Invalid peer Msgs | 0 | 0 | 0 |
| Protocol errors | 0 | 0 | 0 |
| Transient Failures | 0 | 0 | 0 |
| Permanent Failures | 0 | 0 | 0 |

Total Accounting Server Diameter msg Stats:

----Lifetime ----

| | Recent | Total | PerMax |
|------------------|--------|-------|--------|
| CER Sent | 0 | 0 | 0 |
| CEA Success | 0 | 0 | 0 |
| CEA Errors | 0 | 0 | 0 |
| ACR Sent | 0 | 0 | 0 |
| ACA Success | 0 | 0 | 0 |
| ACA Errors | 0 | 0 | 0 |
| DWR Sent | 0 | 0 | 0 |
| DWA Success | 0 | 0 | 0 |
| DWA Errors | 0 | 0 | 0 |
| DWR Rcvd | 0 | 0 | 0 |
| DWA Sent Success | 0 | 0 | 0 |
| DWA Sent Errors | 0 | 0 | 0 |

Total Accounting Server SIPD to RADD msg Stats:

----Lifetime ----

| | Recent | Total | PerMax |
|-------------------|--------|-------|--------|
| Acct Type Start | 0 | 0 | 0 |
| Acct Type Stop | 0 | 0 | 0 |
| Acct Type Interim | 0 | 0 | 0 |
| Acct Type Event | 0 | 0 | 0 |

Server 1:

-----10.196.2.7:3869 -----

Connection state: UNAVAIL

Socket FD: -1

Reconnect attempt in 6s

Pending: 0 of 255

Current RTT: 0 ms(before failure)

| Local IPPort | Remote IPPort | Socket State |
|-----------------|-----------------|--------------|
| 10.196.165.97:0 | 10.196.2.7:3869 | INITIAL |

Server 2:

-----10.196.165.86:3868 -----

Connection state: UNAVAIL

Socket FD: -1

Reconnect attempt in 18s

Pending: 0 of 255

Current RTT: 0 ms(before failure)

| Local IPPort | Remote IPPort | Socket State |
|-----------------|--------------------|--------------|
| 10.196.165.97:0 | 10.196.165.86:3868 | INITIAL |

The system also displays DPR and DPA traffic on a server-specific basis within the **show accounting all**, **show accounting connections**, and **show accounting <IPPort>** commands.

```
DPR Sent          0          0          0
DPA Success       0          0          0
DPA Errors        0          0          0
DPR Rcvd          0          0          0
DPA Sent Success  0          0          0
DPA Sent Errors   0          0          0
```

You use the **servers** argument to display those servers resolved by DNS, based on whether they reside in the primary or secondary pool. This command displays the servers based on IP resolution and includes DNS detail including priority and weightage.

```
SBC01# show accounting servers
07:23:40-198
FQDN: primary.oracle.com
Type: Primary
Last Resolved TimeStamp: Nov 16 08:31:01.521
Min TTL : 120 secs

Server 1:
-----10.196.2.7:3869 -----
Connection state: UNAVAIL
Socket FD: -1
Reconnect attempt in 6s
Pending: 0 of 255
Current RTT: 0 ms(before failure)
Priority: 1
Weightage: 10
TTL: 120 secs
OutStanding ACR:10
Is Marked for Delete : True/False
   Local IPPort      Remote IPPort      Socket State
   10.196.165.97:0   10.196.2.7:3869   INITIAL

Server 2:
-----10.196.165.86:3868 -----
Connection state: UNAVAIL
Socket FD:-1
Reconnect attempt in 18s
Pending: 0 of 255
Current RTT: 0 ms(before failure)
Priority: 1
Weightage: 10
TTL: 120 secs
OutStanding ACR: 10
Is Marked for Delete : True/False
   Local IPPort      Remote IPPort      Socket State
   10.196.165.97:0   10.196.165.86:3868  INITIAL

FQDN: secondary.oracle.com
Type: Secondary
Last Resolved TimeStamp: Nov 16 08:31:01.521
```

```

Min TTL : 120 secs

Server 1:
-----10.196.2.7:3869 -----
Connection state: UNAVAIL
Socket FD: -1
Reconnect attempt in 6s
Pending: 0 of 255
Current RTT: 0 ms(before failure)
Priority: 1
Weightage: 10
TTL: 120 secs
OutStanding ACR: 10
Is Marked for Delete : True/False
   Local IPPort      Remote IPPort      Socket State
   10.196.165.97:0   10.196.2.7:3869   INITIAL

Server 2:
-----10.196.165.86:3868 -----
Connection state: UNAVAIL
Socket FD: -1
Reconnect attempt in 18s
Pending: 0 of 255
Current RTT: 0 ms(before failure)
Priority: 1
Weightage: 10
TTL: 120 secs
OutStanding ACR: 10
Is Marked for Delete : True/False
   Local IPPort      Remote IPPort      Socket State
   10.196.165.97:0   10.196.165.86:3868   INITIAL

```

When using DNS to resolve CCF servers, you can use the following additional commands can help you determine DNS status, traffic and resolutions, within the context of the DNS server operating as an Edge Application Server (EAS)

- show dns cache-entry-eas
- show dns stats-eas
- reset dns-eas

Configuring Diameter-based Accounting

Diameter-based Rf accounting relies on many of the same configuration elements used for RADIUS based accounting. The following two sections explain how to configure both the **account-config** and **account-servers** configuration elements. In addition, you must ensure that accounting is enabled for each realm where you want it to occur. The **accounting-enable** parameter in the realm-config is enabled by default.

Configure the Global Diameter-based Accounting (Rf) Features

To configure the global Diameter-based accounting (Rf) features in the account-config:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ORACLE# configure terminal
```

2. Type **session-router** and press Enter.

```
ORACLE(configure)# session-router
```

3. Type **account-config** and press Enter.

```
ORACLE(session-router)# account-config  
ORACLE(account-config)#
```

4. **hostname**—The hostname for this system. This value must be set to “localhost” or the accounting configuration does not work properly.
5. **port**—Enter 3868 for the RFC-recommended Diameter port number. You may enter a different port number.
 - minimum: 1025
 - maximum: 65535
6. **strategy**—Set the strategy used to select the accounting server which the SBC sends accounting messages. The following are the available strategies:
 - **hunt**—Selects accounting servers in the order in which they are listed. If the first accounting server is online, working, and has not exceeded any of the defined constraints, all traffic is sent to it. Otherwise the second accounting server is selected. If the first and second accounting servers are offline or exceed any defined constraints, the third accounting server is selected. And so on through the entire list of configured servers
 - **failover**—Uses the first server in the list of predefined accounting servers until a failure is received from that server. Once a failure is received, it moves to the second accounting server in the list until a failure is received. And so on through the entire list of configured servers.
 - **round robin**—Selects each accounting server in order, distributing the selection of each accounting server evenly over time.
 - **fastest round trip time**—Selects the accounting server that has the fastest round trip time (RTT) observed during transactions with the servers (sending a record and receiving an ACK).
 - **fewest pending**—Selects the accounting server that has the fewest number of unacknowledged accounting messages (that are in transit to the SBC).
7. **protocol**—Set this parameter to **diameter** to use the Rf accounting interface with a Diameter-based accounting server.
8. **state**— Enter **enabled** to use accounting on this system.
9. **dns-realm**—If using FQDN lookup of account-servers, specify the realm through which the system reaches DNS resources to resolve your FQDN(s) to IP address(es).
10. **max-msg-delay**—Retain the default value of **60** seconds or indicate the length of time in seconds that you want the SBC to continue trying to send each accounting message. During this delay, the SBC can hold a generic queue of 4096 messages.
 - Minimum: zero (0)
 - Maximum: 4294967295

- 11. max-wait-failover**—Retain the default value of **100** messages or indicate the maximum number of accounting messages the SBC can store its message waiting queue for a specific accounting server, before it is considered a failover situation.
- Once this value is exceeded, the SBC attempts to send its accounting messages, including its pending messages, to the next accounting server in its configured list.
- Minimum: one (1) message
 - Maximum: 4096 messages
- 12. trans-at-close**—Retain the default value of **disabled** if you do not want to defer the transmission of message information to the close of a session. Enter **enabled** if you want to defer message transmission.
- **disabled**—The SBC transmits accounting information at the start of a session (Start), during the session (Interim), and at the close of a session (Stop). The transmitted accounting information for a single session might span a period of hours and be spread out among different storage files.
 - **enabled**—Limits the number of files on the SBC used to store the accounting message information for one session. It is easiest to store the accounting information from a single session in a single storage file.
- 13. generate-start**—Retain the default value **ok** if you want the ACR Start message to be generated once the SBC receives an OK message in response to an INVITE.

Other options include:

- **none**—Accounting Start message should not be generated.
 - **invite**—Accounting Start message should be generated once the SBC receives a SIP INVITE.
 - **""**—When two quotation marks are entered next to each other (empty), behavior is identical to none value.
- 14. generate-interim**—Retain the default value **reinvite-response** to cause the SBC to send an Interim charging message to the accounting server.
- You can select none, one, or more than one of the following values:
- **ok**—Start message is generated when the SBC receives an OK message in response to an INVITE.
 - **reinvite**—Interim message is generated when the SBC receives a SIP session reINVITE message.
 - **reinvite-response (default)**—Interim message is generated when the SBC receives a SIP session reINVITE and responds to it (for example, session connection or failure).
 - **reinvite-cancel**—Interim message is generated when the SBC receives a SIP session reINVITE, and the Reinvite is cancelled before the SBC responds to it.
 - **unsuccessful-attempt**—Interim message is generated when a SIP session set-up attempt from a preference-ordered list of next-hop destinations is unsuccessful. This can happen when a local policy lookup, LRT lookup, ENUM query response, or SIP redirect returns a preference-ordered list of next-hop destinations. The interim message contains: the destination IP address, the disconnect reason, a timestamp for the failure, and the number that was called.
- 15. generate-event**—Enter one or more valid events that prompt creation of an Event record. Enclosed multiple values in parenthesis and separate by spaces.

value settings include triggering the SBC to generate an accounting message when it:

- **none**—(default)

- Register—Receives a SIP REGISTER.
 - Local-Register—Prompts the SBC to create Event ACRs on a re-REGISTER to which it replied.
 - message—Receives an SMS message.
16. **intermediate-period**—Enter amount of time in seconds between generating periodic interim ACR messages during a SIP call. This parameter defaults to zero, which disables continuous Interim charging messages.
 17. **vsd-id-range**—Ensure that this parameter is left blank when communicating with a Diameter-based Rf accounting server.
 18. **max-acr-retries** — Retain the default value of zero (0) or enter the maximum number of times that the SBC can resend an ACR for a session.
 - Minimum: zero (0)
 - Maximum: 4
 19. **acr-retry-interval** — Retain the default value of 10 seconds or enter the time in seconds for the SBC to wait before resending an ACR for a session.
 - Minimum: 5
 - Maximum: 20
 20. **acr-buffer-upper-threshold**—The upper threshold for the ACR buffer after which the SBC will select an alternate server.
 - 90 (default)
 - 0 - 100
 21. **acr-buffer-lower-threshold**—The lower threshold for the ACR buffer which, when reached, the SBC will select the primary server again.
 - 70 (default)
 - 0 - 100
 22. **maintain-ccf-affinity**—Enable affinity between ACRs and CCFs so that all ACRs within a single session are sent to the same CCF, unless it goes down.
 - Disable (default)
 - Enable
 23. **send-disconnect-peer-msg**— Enables the system to send a DPR to the CCF before disconnecting from the CCF
 - Disable (default)
 - Enable
 24. **next-priority-selection-interval**—Specifies in minutes how long the system waits after experiencing an overload scenario before it refers to the next priority. The range is 0 to 60 minutes.
 25. Save your work.

Configure Accounting Servers

You can create one or more servers to which the SBC can send accounting messages. If using FQDNs in the hostname parameters, you can create a maximum of two **account-servers**.

1. Continuing from the previous account-config configuration, enter the account server sub-element by typing **account-servers** Enter.

```
ORACLE(account-config)# account-servers
ORACLE(account-server)#
```

2. **hostname**: Enter the hostname of the accounting server. This entry can be an IPv4 Address for RADIUS servers, or an FQDN, IPv4 or IPv6 for Diameter servers.
3. **fqdn-pooltype**: When accessing a diameter accounting server using an FQDN, specify whether the pool of servers resolved to this FQDN is the primary or secondary pool of servers with which the system performs accounting functions. If all servers resolved to the primary pool are unavailable, the system uses the secondary pool from which it accesses CCF servers.
4. **port**: Enter 3868 for the RFC-recommended Diameter port number. You may enter a different port number if desired.
 - minimum: 1025
 - maximum: 65535
5. **state**: Retain the default enabled to enable this account server or enter disabled to disable it.
6. **min-round-trip**: Retain the default 250 milliseconds or indicate the minimum round trip time of an accounting message.
 - minimum: 10 milliseconds
 - maximum: 5000 milliseconds

A round trip consists of the following:

The SBC sends an accounting message to the account server.

The account server processes this message and responds back to the SBC.

If the fastest RTT is the strategy for the account configuration, the value you enter here can be used to determine an order of preference (if all the configured account servers are responding in less than their minimum RTT).

7. **max-inactivity**: Retain the default 60 seconds or indicate the length of time in seconds that you want the SBC with pending accounting messages to wait when it has not received a valid response from the target account server.
 - minimum: 1 second
 - maximum: 300 seconds

Once this timer value is exceeded, the SBC marks the unresponsive account server as disabled in its failover scheme. When a server connection is marked as inactive, the SBC attempts to restart the connection and transfers pending messages to another queue for transmission. Accounting messages might be moved between different account servers as servers become inactive or disabled.

8. **restart-delay**: Retain the default 30 seconds or indicate the length of time in seconds you want the SBC to wait before resending messages to a disabled account server.
 - minimum: 1 second
 - maximum: 300 seconds
9. **priority**: Enter the number corresponding to the priority of this account server, for use with server prioritization. The default for this parameter is 0, meaning the prioritization feature is

turned off: and that the SBC will therefore prioritize accounting servers by IP address and port.

10. **origin-realm:** Enter the value of the Diameter AVP Origin-Realm (code 296) to include in Accounting-Request (ACR) messages sent by the SBC to the Rf accounting server. If you do not enter a value, AVP 296 is not populated.
11. **domain-name-suffix:** Enter the suffix to be appended to any Diameter FQDN or Diameter Identity used when the SBC communicates with the Diameter Rf accounting server. Your value can be any string, to which the SBC will prepend with a dot.
12. **dns-query-type:** Enter the type of query you want the system to perform. Applies when **hostname** is set to **FQDN**.
 - None (default): The system refers to the interface associated with the dns-realm setting in the account-config. If the interface has an IPv4 address, the system performs an A query; If the interface has an IPv6 address, the system performs an AAAA query.
 - A: The system performs an A query.
 - AAAA: The system performs an AAAA query.
13. **watchdog-ka-timer:** Set this parameter to the value in seconds that the SBC waits between sending DWRs. 0 disables this feature. Valid non-zero values are 6 - 65535
14. Save your work.

Create a Dictionary File for Decoding AVPs

You can generate an AVP dictionary from the SBC to install and use for decoding Oracle-specific Rf AVPs in messages using Wireshark. After generating this dictionary, you include it within your Wireshark deployment and configure a Wireshark resource file. This allows Wireshark to decode standalone and grouped AVPs identified with the ACME_DIAM_VENDOR_ID label.

To create this dictionary, you:

1. Login to your cloud SBC.
2. Run the command **dump_diam_dict** from the ACLI. The system creates the file `/opt/logs/OracleSBCRf.xml`.
3. Copy the file from the SBC using SFTP.

Having generated the file, you next establish it as a Wireshark decode entity.
4. Navigate to the path where your Wireshark application is installed. On Windows, this is typically "C:\Program Files\Wireshark\diameter".
5. Open the dictionary.xml file, using a notepad application with Admin privileges. This file contains an assortment of content types, one of which is called an "ENTITY". The ENTITY list allows Wireshark to use decoding resources.
6. Add a new entry to the ENTITY list, using the name of your AVP file. This procedure uses the example file name OracleSBCRf.xml. The code block below gives an example of two items in the ENTITY list, one of which is your new entry.

```
...
<!ENTITY Custom SYSTEM "Custom.xml">
<!ENTITY OracleSBCRF SYSTEM "OracleSBCRF.xml">
...
```

7. Close this file using the following syntax at the end of the file.

```
...
&Custom;
&OracleSBCRF;
</dictionary>
```

8. Put your OracleSBCRF.xml file in the same directory as the "diameter" file.
9. Start or restart Wireshark.

At this point, Wireshark can decode the applicable AVPs sent within Rf interface messages.

Additional Rf Features

This section describes additional features that you can configure for Diameter accounting, as well as relevant alarm and SNMP trap behavior.

Service-Context-ID Format

The Service-Context-ID AVP (461) located in the root ACR message is formatted as follows:

```
[[["extensions".]MNC.MCC.]"Release".]32260@3gpp.org
```

where

- extensions—This is operator specific information to any extensions in a service specific document. The value is configured by setting the **diam-srv-ctx-ext** parameter.
- MNC.MCC—This identifies the operator implementing the service specific document, which is used to determine the specific requirements for the operator configurable parameters. Both MNC and MCC must be specified separated by a dot(.). The value is configured by setting the **diam-srv-ctx-mnc-mcc** parameter as two integers separated by a dot. For example: 012.310
- Release—This indicates the 3GPP Release the service specific document is based upon e.g. 6 for Release 6. The value is configured by setting the **diam-srv-ctx-rel** parameter with valid values are >=1.

Note

"32260@3gpp.org" is fixed.

Acme Excluded Attribute Range

You can select certain ACME specific AVPs to include in the Rf accounting records with the **diam-acme-attr-id-range** parameter. If this parameter is configured with one or more values, then all other valid Acme-specific AVPs, by number, are excluded. If by configuration, the SBC will exclude one (or more) individual ACME attributes, there will be no effect. If by configuration an Acme-specific attribute number that refers to a group is excluded, the SBC removes the complete grouped AVP from the ACR message.

Consider:

- Acme-specific attribute 1—The grouped AVP

- Acme-specific attributes 2-35—The individual AVPs that make up the group

If you configure **diam-acme-attr-id-range 1,3-** the SBC includes all attributes in the Acme group; This configuration aims to exclude only attribute 3 but is has no effect.

If you configure **diam-acme-attr-id-range 2-**, the SBC excludes the full Acme-specific group because Acme-Packet-Specific-Extension-Rf AVP (1) was not included.

The **diam-acme-attr-id-range** parameter's syntax is as follows:

| Syntax | Meaning |
|--------|---|
| X-Y | include range of attribute IDs from X to Y (X and Y are included) |
| -Y | include any attribute ID <= Y |
| X- | include any attribute ID >= X |
| - | include any attribute ID |
| X | include attribute ID = X |

Configure Account

1. In Superuser mode, type **configure terminal** and press Enter.

```
ORACLE# configure terminal
```

2. Type **session-router** and press Enter.

```
ORACLE(configure)# session-router
```

3. Type **account-config** and press Enter.

```
ORACLE(session-router)# account-config
ORACLE(account-config)#
```

4. **diam-srv-ctx-ext**—Enter the extension portion of the Service-Context-ID AVP value. This value can be any string.
5. **diam-srv-ctx-mnc-mcc**—Enter the MNC.MCC portion of the Service-Context-ID AVP value. This value must follow the NUM1.NUM2 format.
6. **diam-srv-ctx-rel**—Enter the release portion of the Service-Context-ID AVP value. This value can be any number >= 1..
7. **diam-acme-attr-id-range**—Enter the range of Acme-specific AVPs to include in ACR messages. Leaving this parameter blank or configured with a - includes all AVPs.
8. Type **done** to save your work.

Including the To Header in ACRs and CDRs

You can configure the SBC to support the Acme-SipHdr-TO AVP. This AVP conveys the value of TO headers in Rf deployments. The system uses this AVP to populate the string in the sipHdrTO from SIP methods into ACRs and CDRs. Enabling this feature causes the system record the SIP 'To:' header in ACRs for all endpoints.

When configured, the SBC uses this AVP (code 122), to capture TO headers from incoming SIP message and populate the Acme-SipHdr-TO in the Acme-Packet-Specific-Extension-Rf grouped AVP.

By default, the SBC does not populate this AVP. You configure the **sip-tohdr-in-acr** option in the **account-config** to enable this behavior.

```
ORACLE(account-config)#options +sip-tohdr-in-acr
```

Supporting IOI AVPs for Unregistered Endpoints

You can configure the SBC to include the Originating-IOI and Terminating-IOI AVPs within ACRs and Diameter based CDRs for unregistered endpoints in addition to registered endpoints. Support for registered endpoints is available without special configuration. For unregistered endpoints, you enable the **ioi-for-unregistered** option within the **account-config** element.

The Inter-Operator-Identifier AVP (838) is a grouped AVP that includes the originating IOI AVP (839) and the terminating IOI AVP (840) for the purpose of tracking inter-service provider traffic. The SBC extracts this information from incoming P-Charging-Vector headers in the 200 OK of an initial INVITE from either side of a peering deployment. When applicable, the SBC populates these AVPs within:

- **START ACR**—Requires that you configure the **generate-start** parameter to **OK** in the **account-config**.
- **INTERIM ACR**
- **STOP ACR**
- **EVENT ACR**

If there is no IOI info in the P-Charging-Vector, the SBC does not include these AVPs in ACRs.

You configure the **charging-vector-mode** to **pass** to convey these AVPs for unregistered endpoints. If you also enable the **realm-as-ioi** option, however, the SBC ignores the value derived from the **pass** parameter and includes the ingress and egress realm names in the AVPs.

The table below presents the orig-ioi and term-ioi that the SBC would send in its ACRs based on your **realm-as-ioi** and **ioi-for-unregistered** options configurations. Assume the ingress and egress realm names are realm1 and realm2. For the bottom three rows, assume the following P-Charging-Vector contents.

```
P-Charging-Vector: icid-value=89000078phcsioo6ohb4jq46e9c6nk1zz9e432-6;
icid-generated-at=171.15.252.1;orig-ioi=AAAA;term-ioi=BBBB
```

| realm-as-ioi | ioi-for-unregistered | IOI in ACR for unregistered endpoints | Orig-ioi, term_ioi in ACR |
|---------------------|-----------------------------|--|----------------------------------|
| Disabled | Disabled | Not Present | Absent, Absent |
| Disabled | Enabled | From the PCV in 200OK, if all conditions fulfilled | AAAA, BBBB |
| Enabled | Disabled | Ingress and Egress realm names | realm1, realm2 |
| Enabled | Enabled | Ingress and Egress realm names | realm1, realm2 |

You configure the **ioi-for-unregistered** option in the **account-config** to include the IOI AVPs in Diameter CDRs for unregistered endstations. If this option is not set, the SBC adds IOI AVPs in Diameter CDRs for registered endstations only.

Configure the **ioi-for-unregistered** option using the syntax below.

```
ORACLE(account-config)# options +ioi-for-unregistered
```

If you type options and then the option value without the plus sign, you overwrite any previously configured options. To add a new option to an options list, prepend the new option with a plus sign as shown in the previous example.

Managing IOI for Peering Endpoints

You can configure the SBC to manage P-Charging-Vector (PCV) headers and populate standard AVPs and CDRs with Originating and Terminating IOIs for peering deployments. This is separate from IOI management within access and P-CSCF applications. Configurations that apply to this feature include the **ioi-for-unregistered** option in the applicable **account-config**, and the **charging-vector-mode** settings on the applicable **sip-interface** elements.

This feature allows the SBC to do the following for peering applications, wherein endpoints do not register with the SBC:

- Manage the inclusion and contents of PCV headers with respect to the IOI and ICID parameters in INVITEs, 18x and 2xx messages.
When you configure the **ioi_for_unregistered** option, the SBC refers to IOIs from SIP messages to generate ICID values. If the SIP message doesn't contain the PCV header from the access side, however:
 - The SBC generates and inserts an ICID.
 - The SBC does not insert IOIs in SIP messages or ACRs.
- Include IOI AVPS 838, 839, and 840 in START, INTERIM, STOP, and EVENT ACRs and subsequent CDRs for the applicable end points.
When you configure the **charging-vector-mode** on the access side **sip-interface** to **insert_ioi_pass** in conjunction with the **ioi_for_unregistered** option, the SBC inserts IOIs from the ingress SIP message in the applicable CDRs. In addition, configure the **charging-vector-mode** on the core side **sip-interface** to **delete-and-respond**. If available, uses the IOI received in the applicable SIP messages. The SBC inserts either Orig or Term IOIs, depending on the call flow.

The SBC sends the same values for the IMS-Charging-Identifier (AVP Code: 841) and Inter-Operator-Identifier (AVP Code: 838) AVPs in ACRs.

Caveats to consider when using this feature include:

- The feature is applicable to Peering deployments.
- The feature also supports registered calls, when setting the ingress interface **charging-vector-mode** to **insert_ioi_pass**.
- An IPX Hub does not change the IOI (orig / term) during a call. This means the SBC only needs to consider the initial IOI, which is received in an originating INVITE or 18x / 200 OK messages (terminating).
- Refer, Reinvite, Update, SMS, Register scenarios are not in the scope of this feature.
- This feature does not apply to SIP forking call flows.

Configuring this Feature

You configure this feature by enabling the **ioi-for-unregistered** option in the applicable **account-config**. The code block below depicts this configuration.

```
ORACLE(account-config)# options +ioi-for-unregistered
```

In addition, you set the **charging-vector-mode** on the access **sip-interface** to **insert_ioi_pass**, and the **charging-vector-mode** on the core **sip-interface** to **delete-and-respond**. The code block below depicts these configurations.

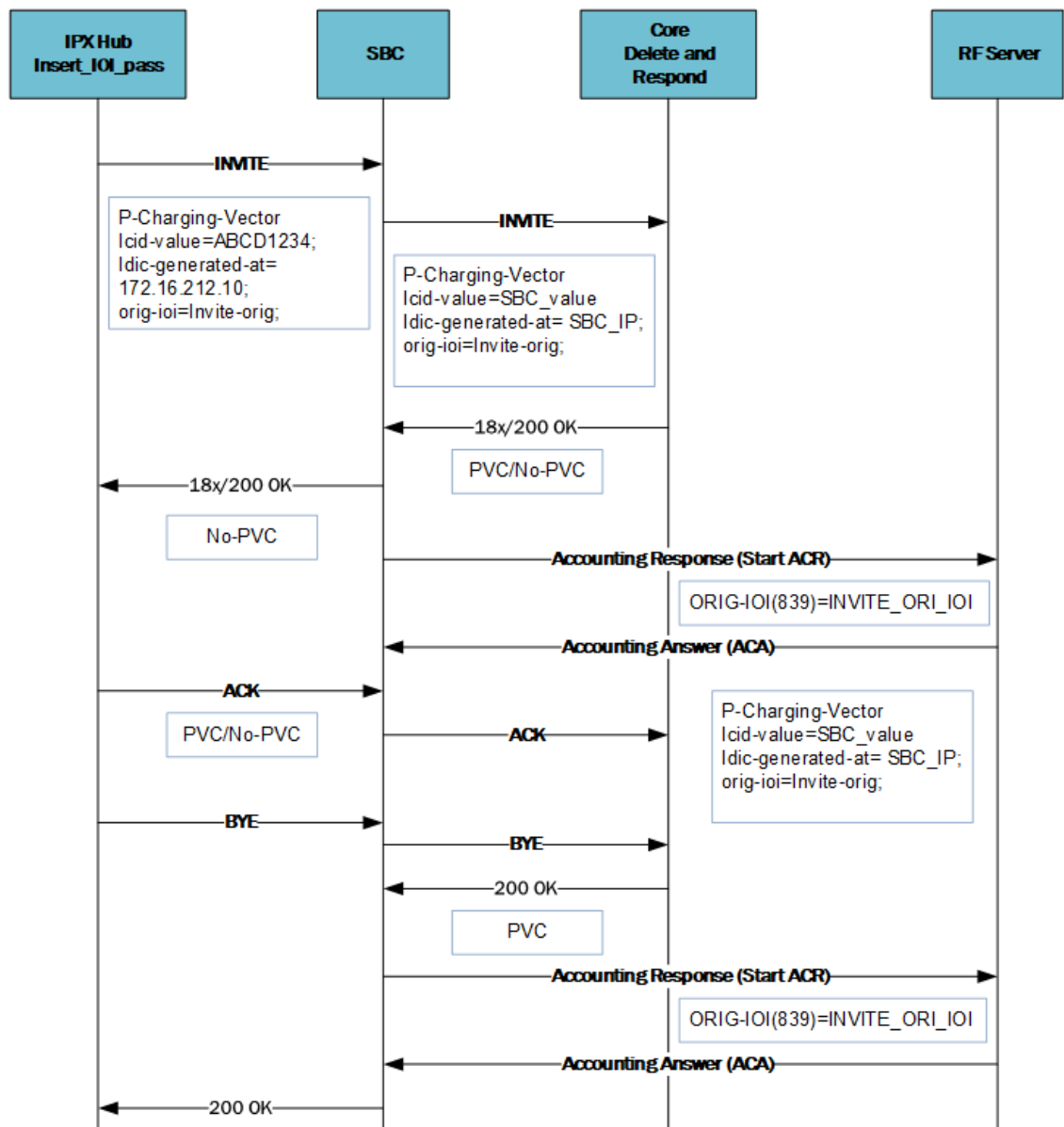
```
sip-interface
    realm-id                CoreRealm
    sip-port
...
    operator-identifier      Core_Side
    charging-vector-mode     delete-and-respond
sip-interface
    realm-id                AccessRealm
    sip-port
...
    operator-identifier      IPX_HUB
    charging-vector-mode     insert_ioi_pass
```

The **realm-as-ioi** feature supersedes this feature. If you have configured the **realm-as-ioi** option, the SBC does not transmit the PCV from the 200 OK to the Rf server. Instead, the SBC sends realm names as IOIs for accounting in diameter, radius and local CDRs.

Originating IOI Call Flow

Conditions for the access side call (IPX Hub) terminating at the Core side, depicted in the diagram below, include:

1. The call is initiated from the IPX Hub side.
2. You have configured the **charging-vector-mode** on the caller's **sip-interface** to **insert_ioi_pass**.
3. The Orig_IOI present in the SIP P-Charging-Vector (PCV) header within the ingress INVITE.



Processing for this call includes:

- In the egress Invite, while populating the P-Charging-Vector, the SBC uses the IOI from the ingress INVITE and generates the ICID values.
- The **icid-generated-at** parameter in the PCV is the IP address of the SBC **sip-interface**.
- The SBC replaces the ICID values received in the Ingress Invite with its own generated values.
- Because there are no IOI values provided from the core side (**delete-and-respond**), this scenario does not include any IOI values in the 18x/2xx messages.
- In this scenario, the SBC includes only the Originating-IOI in the ACR towards the Rf server.

Note

The SBC handles any PCV parameters coming from the core side according to the configured **charging-vector-mode**.

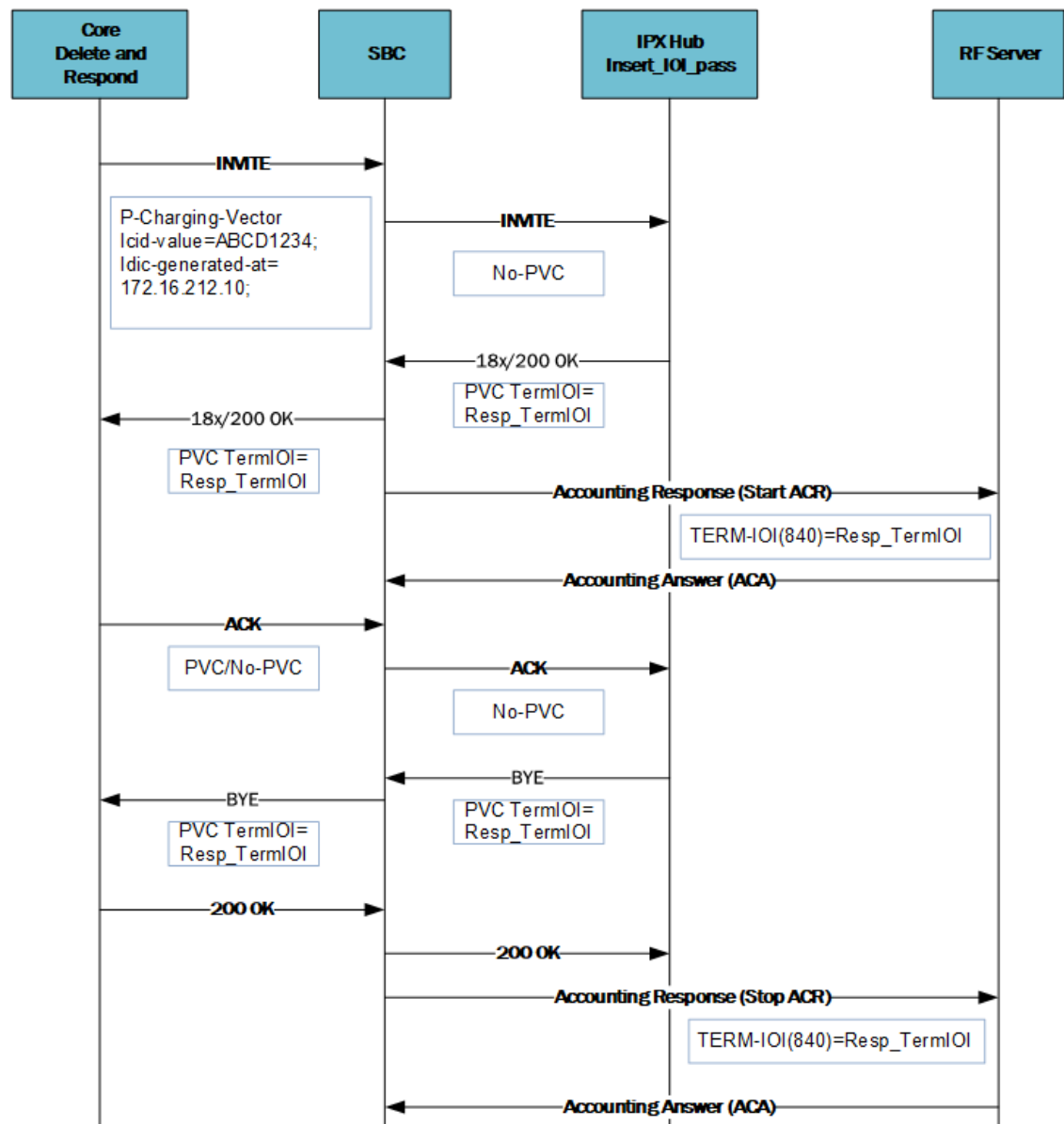
Note

The SBC uses the initial/first IOI in the ACR towards the Rf server.

Terminating IOI Call Flow

Conditions for the core side call terminating at the access side, depicted in the diagram below, include:

1. The call is initiated from the core side.
2. You have configured the **charging-vector-mode** on the caller's **sip-interface** to **delete_and_respond** causing the SBC to drop the PCV.
3. The ingress 18x/2xx has the Term-IOI, which the SBC uses to populate the ACR and the egress response towards the core along with the ICID values from Initial INVITE.



Processing for this call includes:

- The SBC does not include a PCV in the egress INVITE.
- The SBC inserts the stored PCV (icid-values) from Delete and Respond charging-vector-mode in the egress response.
- The SBC inserts the stored term-IOI from the ingress response (18x/2xx) in the egress response.
- The SBC includes only the Terminating-IOI from the 18x/2xx Term IOI values in the ACR towards the Rf server.

Including Number Portability Routing Information in ACRs and CDRs

You can configure the SBC to support the Number-Portability-Routing-Information AVP, which contains the routing number (**rn**) received from a mobile number portability (MNP) request. This AVP is grouped in the IMS-Information group AVP.

When support for this AVP is configured and **rn** appears in the URI header of the initial SIP INVITE or the MNP response from the ENUM server (with either a SIP or TEL URI), the SBC does the following:

1. Extracts the **rn** value from the Request URI header in the SIP INVITE or ENUM response.
2. Adds the **rn** to the Request URI header of the outgoing INVITE.
3. Populates the Number-Portability-Routing-Information AVP in the ACR with the **rn** value.

If **rn** appears in both the initial INVITE and the ENUM SIP response, the value from the ENUM takes priority.

If no **rn** appears in the ENUM SIP response, the SBC checks for it in request URI header of the initial INVITE.

The AVP is not populated in the following situations:

- SIP: **rn** does not appear in either the initial INVITE or the ENUM response
- TEL: **rn** does not appear in the ENUM response. The initial TEL INVITE will never include **rn**, so the SBC does not check there.
- The ENUM server sends no response or an error response and **rn** does not appear in the initial INVITE.
- By default, when the **number-port-routing-infn-avp** option in the **account-config** configuration element is not set.

Generally, the AVP is included in START, STOP, and INTERIM records. However, the data depends on ENUM response data, which is not returned until after the initial INVITE. When the **generate-start** parameter of **account-config** is set to **invite**, the AVP cannot be populated for START records.

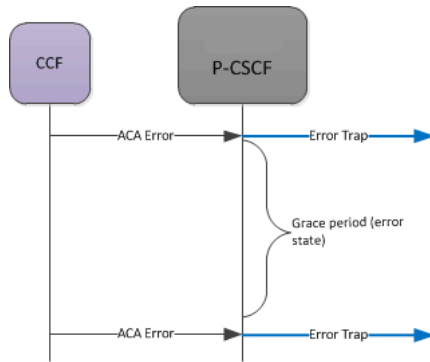
You enable the SBC to populate this AVP by configuring the **number-port-routing-infn-avp** option in **account-config**:

```
ORACLE(account-config)#options +number-port-routing-infn-avp
```

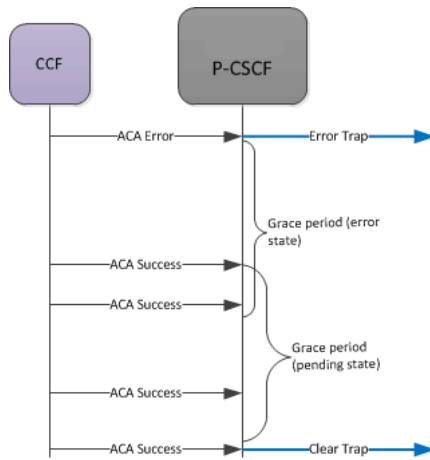
SNMP Trap Behavior

The SBC sends an SNMP trap (apDiameterSvrErrorResult) upon a CCF returning an error-containing ACA. See the list of four errors (3002, 3004, 4002, 5012) which generate traps in the [Alarms](#) section. The frequency at which subsequent traps are sent is based upon configuring the **diam result code trap grade period** option configured in the account config.

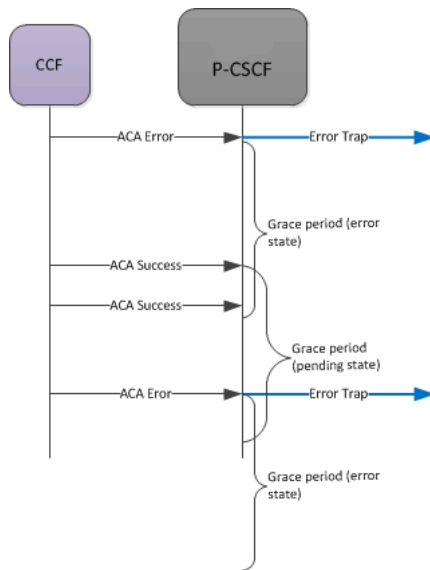
When the SBC has sent a trap after receiving a bad ACA, it goes into an error state. The SBC waits one grace period before checking if it is still in an error state. If the state has not switched from errored back to pending, the SBC sends another error trap, after that first grace period ends (counting from the initial error) and then after the next error message is received.



If the CCF returns a successful message, the grace start in a pending state. When this second timer expires (pending no more errors or additional successes), on the next successful ACA, a success trap is sent.



If, while in the pending grace period an ACA error is received, the SBC immediately sends an error trap, and begins the error state again. It also starts counting the initial grace period time again.



Alarms

A MINOR non health affecting Diameter Accounting Server Error alarm will be generated when one of the following Result Codes is received:

- 3002 (DIAMETER_UNABLE_TO_DELIVER)
- 3004 (DIAMETER_TOO_BUSY)
- 4002 (DIAMETER_OUT_OF_SPACE)
- 5012 (DIAMETER_UNABLE_TO_COMPLY)

The alarm is cleared when a success (2XXX) code is received.

The rules for setting state of the failed server alarm are the same as the grace period rules described in the [SNMP Trap Behavior](#) section.

For example:

```
327703 835778540      5      2012-03-13 13:03:34      2012-03-13 13:03:34
Count  Description
1      Diameter Accounting Server Returned Error Result Code|
172.30.0.135:3869-5012|172.30.69.211:3868-3002
```

SNMP MIBs and Traps

ApDiamResultCode Textual Convention

```
ApDiamResultCode ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "The Result-Code AVP (268) value
        RFC 3588, 7.1. Result-Code AVP"
    SYNTAX          INTEGER {
        diameterMultiRoundAuth(1001),
        diameterSuccess(2001),
        diameterLimitedSuccess(2002),
        diameterCommandUnsupported(3001),
        diameterUnableToDeliver(3002),
        diameterRealmNotServed(3003),
        diameterTooBusy(3004),
        diameterLoopDetected(3005),
        diameterRedirectIndicatoion(3006),
        diameterApplicationUnsupported(3007),
        diameterInvalidHdrBits(3008),
        diameterInvalidAvpBits(3009),
        diameterUnknownPeer(3010),
        diameterAuthenticationRejected(4001),
        diameterOutOfSpace(4002),
        electionLost(4003),
        diameterAvpUnsupported(5001),
        diameterUnknownSessionId(5002),
        diameterAuthoriszationRejected(5003),
        diameterInvalidAvpValue(5004),
```

```

        diameterMissingAvp(5005),
        diameterResourcesExceeded(5006),
        diameterContradictingAvps(5007),
        diameterAvpNotAllowed(5008),
        diameterAvpTooManyTimes(5009),
        diameterNoCommonApplication(5010),
        diameterUnsupportedVersion(5011),
        diameterUnableToComply(5012),
        diameterInvalidBitInHeader(5013),
        diameterInvalidAvpLength(5014),
        diameterInvalidMessageLength(5015),
        diameterInvalidAvpBitCombo(5016),
        diameterNoCommonSecurity(5017)
    }
}

```

apDiameterSrvrErrorResultTrap

```

apDiameterSrvrErrorResultTrap      NOTIFICATION-TYPE
OBJECTS                            { apDiamAcctSrvrHostName,
                                   apDiamAcctSrvrIPPort,
                                   apDiamAcctSrvrOriginRealm,
                                   apDiamAcctSrvrOriginHost,
                                   apDiamAcctSrvrTransportType,
                                   apDiameterResultCode
                                   }
STATUS                              current
DESCRIPTION
    " The trap can be generated when the Diameter Server
      returns 3xxx (Protocol Errors), 4xxx (Transient Failures), or
      5xxx (Permanent Failure) Result-Code AVP (268)"
 ::= { apDiamNotifications 5 }

```

apDiameterSrvrSuccessResultTrap

```

apDiameterSrvrSuccessResultTrap    NOTIFICATION-TYPE
OBJECTS                            { apDiamAcctSrvrHostName,
                                   apDiamAcctSrvrIPPort,
                                   apDiamAcctSrvrOriginRealm,
                                   apDiamAcctSrvrOriginHost,
                                   apDiamAcctSrvrTransportType,
                                   apDiameterResultCode
                                   }
STATUS                              current
DESCRIPTION
    " The trap can be generated when the Diameter Server
      returns a 2xxx (Success) Result-Code AVP (268)
      after an error result"
 ::= { apDiamNotifications 6 }

```

apDiamACCTResultObjectsGroup Object Group

```

apDiamACCTResultObjectsGroup OBJECT-GROUP
  OBJECTS {
    apDiameterResultCode
  }
  STATUS      current
  DESCRIPTION
  "A collection of mib objects accessible only to traps."
  ::= { apDiamNotificationGroups 3 }

```

apDiamACCTResultNotificationsGroup Notification Group

```

apDiamACCTResultNotificationsGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    apDiameterSrvrErrorResultTrap,
    apDiameterSrvrSuccessResultTrap
  }
  STATUS      current
  DESCRIPTION
  "A collection of traps defined for ACCT Result Code."
  ::= { apDiamNotificationGroups 4 }

```

SNMP Varbind Definitions

- apDiamAcctSrvrHostName—contains the account-server hostname.
- apDiamAcctSrvrIPPort—This object contains the account-server IP address and port number in the following format:
XXX.XXX.XXX.XXX:PORT
- apDiamAcctSrvrOriginRealm—contains the origin realm, which is a concatenation of the account-server realm and suffix in the following format:
[account-server realm][account-server suffix]
- apDiamAcctSrvrOriginHostName—contains the origin host name, which is a concatenation of the accounting-config host name, account-server realm and account-server suffix in the following format:
[accounting-config host name].[account-server realm][account-server suffix]
- apDiamAcctSrvrTransportType—contains the transport type.
- apDiameterResultCode—contains the Result-Code AVP (268) value as defined in RFC 3588, 7.1. Result-Code AVP

Diameter Rf Charging Buffering and Storage

About Buffering

Diameter Rf Charging, Buffering, and Storage enables the SBC to buffer all accounting requests (ACR) in memory for a configurable number of ACRs. The buffer holds a minimum of 15 minutes of ACRs under busy-hour load conditions. For example, based on intended traffic, the buffer would hold 54 calls-per-second which equals approximately 150,000 records. The

SBC sends an SNMP trap when accounting records begin to drop from the buffer due to an overflow condition. Subsequently, a clearing SNMP trap is sent once the fault condition is removed.

About Storage

The SBC maintains storage to temporarily store accounting records to a charging collection function (CCF) link, in case of a failure or congestion exists on the SBC. In this scenario, the SBC can store a minimum of 3 days-worth of accounting records. All ACRs can remain in storage for a configurable amount of time, and for a minimum of 3 days under normal traffic-load conditions.

There are two configurable options for storing ACRs:

- Store all ACRs generated by the SBC
- No ACRs in storage

Monitoring Storage Space

Disk storage space monitoring can be done on the total drive, or by disk partition. You can monitor storage space using the one of the following methods:

- Command line interface (CLI)
- SNMP management information base (MIB)
- Historical data records (HDR)

ACLI Instructions and Examples

To configure Diameter Rf buffering and storage size:

1. In Superuser mode, type `configure terminal` and press Enter.

```
ORACLE# configure terminal
```

2. Type `session-router` and press Enter.

```
ORACLE(configure)# session-router  
ORACLE(session-router)#
```

3. Type `account-config` and press Enter.

```
ORACLE(session-router)# account-config  
ORACLE(account-config)#
```

- If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI `select` command) the configuration that you want to edit.
4. **diam-attr-id-range**—Comma delimited range of accounting attributes to include in DIAMETER Rf accounting records (blank field means feature turned off and all attributes included).
 5. **msg-queue-size**—Enter the message queue size. **This parameter applies to both RADIUS and Diameter accounting interfaces.** The valid range is 5000 - 150000. The default value is 5000.

6. Save your work.

SNMP

SNMP traps will be sent to the configured management system(s) when accounting records begin to drop due to an overflow condition and when this fault condition is removed:

- apAcctMsgQueueFullTrap will be sent when accounting records begin to drop due to an overflow condition and all accounting servers are down
- apAcctMsgQueueFullClearTrap will be sent when the apAcctMsgQueueFullTrap fault condition is cleared

The following varbinds are defined for the above traps

- apAcctMsgQueueCurrent

The current measured percentage value of space available

- apAcctMsgQueueMinorThreshold

The current configured minor threshold value

- apAcctMsgQueueMajorThreshold

The current configured major threshold value

- apAcctMsgQueueCriticalThreshold

The current configured critical threshold value.

DIAMETER Rf Charging Failure & Recovery Detection

The SBC can detect and report when the DIAMETER Rf interface has failed and when it has recovered.

- Transport failure detection—The SBC sends SNMP traps to the configured management systems when a Diameter Rf Charging transport failure has been detected. If multiple transport failures have been detected, an SNMP trap is sent for each failure.
- Transport recovery detection—When a Diameter Rf Charging CCF has recovered and is back in service, an SNMP trap notification is sent by the SBC to the configured management systems notifying of that event.

Associated Traps

SNMP traps will be sent to the configured management system(s) when a transport failure or recovery is detected:

- apDiameterAcctSrvrDownTrap will be sent if SBC can't connect to a configured Diameter Accounting Server after reboot
- apDiameterAcctSrvrDownTrap will be sent if SBC loses connection to a configured Diameter Accounting Server during normal operations
- apDiameterAcctSrvrUpTrap will be sent if SBC regains connection to a configured Diameter Accounting Server after a previous connection loss

The following varbinds are defined for the above traps:

- apDiamAcctSrvrHostName—This object will contain account-server hostname.

- `apDiamAcctSrvrIPPort`—This object will contain account-server IP address (which is the same as the hostname since we don't support FQDN for the account-server hostname) and port number in the following format: `XXX.XXX.XXX.XXX:PORT`.
- `apDiamAcctSrvrOriginRealm`—This object will contain the origin realm, which is a concatenation of the account-server realm and suffix in the following format:
[account-server realm][account-server suffix]
- `apDiamAcctSrvrOriginHostName`—This object will contain the origin host name, which is a concatenation of the accounting-config host name, account-server realm and account-server suffix in the following format:
[accounting-config host name].[account-server realm][account-server suffix]
- `apDiamAcctSrvrTransportType`—This object will contain the transport type. At this time only the TCP transport type is supported

Custom Diameter ACR AVPs for SIP

When using diameter accounting, you can populate reserved ACR AVPs for SIP calls with custom data by using header manipulation rules (HMR).

You configure HMR to add data to AVP IDs 123 to 153, which are reserved for this purpose. Never use these AVPs for other purposes. You can configure multiple rules to add multiple headers from any SIP message header, which will include multiple custom AVPs in the ACR. See *Header Manipulation Rules Guide* for complete HMR configuration information.

When you configure HMR, the rules apply to RADIUS and diameter accounting, whichever is in use. The reserved AVPs correspond to RADIUS VSAs 200-230. When creating rules, you reference the VSA number, not the AVP number.

For example, the following rules instruct the SBC to locally store the value of the **from** header of a SIP INVITE, then add the **P-Acme-VSA** header with the locally stored value, prefixed with 200. When generating CDRs and ACRs, if a session agent, realm, or SIP interface is configured to use the customVSA ruleset, the SBC checks the SIP message for the **P-Acme-VSA** header, and uses it to add the value to the relevant VSA or AVP; in this case, VSA 200 or AVP ID 123.

```

sip-manipulation
  name          customVSA
  header-rule
    name        storeFrom
    header-name from
    action      store
    comparison-type pattern-rule
    msg-type    request
    methods     INVITE
    match-value .*
  header-rule
    name        generateVSA200
    header-name P-Acme-VSA
    action      add
    methods     INVITE
    new-value   200:+$storeFrom.$0

```

You can configure HMR to populate the custom AVPs for:

- Start ACR: Generated for the initial INVITE (timing depends on the value of **generate-start** on the **account-config**).
- Interim ACR: Generated for reINVITEs (timing depends on the value of **generate-interim** on the **account-config**).
- Stop ACR: Generated when the call is disconnected, after receiving a 200 OK from the callee in response to a BYE.
- Event ACR: Generated for events such as SMS, or 400-level errors.
Custom AVPs are not populated for REGISTER Event ACRs. The Acme-Packet-Specific-Extension-Rf(1) group AVP is not included in the ACR when an event ACRs is generated due to REGISTER.

If HMR configuration is removed during an ongoing call, the custom AVPs are still populated for subsequent ACRs on that call with the values from the last ACR where HMR applied.

For more information about using HMR to populate the custom AVPs, including how to configure the **sip-header-rules** element, see [Custom RADIUS CDR VSAs for SIP](#) in the Call Signaling Accounting Configuration chapter of this guide. The HMR configuration steps are the same for RADIUS and diameter accounting.

Note

In RADIUS accounting, because START records are not created until the session is initiated, you cannot use HMR to add custom VSAs on START records in the CDR. However, because diameter accounting can create START records before the session is initiated, you can use HMR to add custom AVPs on START records in the ACR.

For detailed information about HMR, including functionality, configuration, and examples of other use cases, see *Header Manipulation Rules Guide*.

Logging for Custom Diameter AVPs

The SBC captures logs in the **log.sipd** and **log.radd** files.

The log entries for successful AVP generation appear in the following format:

```
Packing Custom VSAs in ACR recType:      type
Current VSA List size :                   count
  ID = VSA_number VALUE = value
```

where:

- *type* is the ACR type the custom AVPs were added to (**START**, **STOP**, **INTERIM**, or **EVENT**)
- *count* is the total number of custom AVPs added or updated
- *VSA_number* is the VSA, from 200 to 230, that corresponds to the AVP IDs from 123 to 153.

A

RADIUS Dictionary Reference

For RADIUS dictionary content, refer to the radius file in this version's documentation library.

B

Local CDR Table Layouts

This appendix includes example tables that show the fully inclusive layout of local CDR files for the most common CDR records.

You can also create similar layout outputs based on your own configuration and traffic with the **dump_csv_format** command if you use the functions explained in the [RADIUS CDR Content Control](#) section.

Note

The Acme-CDR-Sequence-Number, Vendor ID 59, appears in local CDR files when both file-output is enabled and an account server is configured.

Local CDR Start Record (RADIUS)

The following table contains the local CDR layout, inclusive of all VSA data for a start record, when RADIUS is selected as the account protocol.

Table B-1 Local CDR Start Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------------|--------|------------|
| 1 | Accounting Status | none | 40 |
| 2 | NAS IP Address | none | 4 |
| 3 | NAS Port | none | 5 |
| 4 | Accounting Session ID | none | 44 |
| 5 | Ingress Session ID | ACME | 3 |
| 6 | Egress Session ID | ACME | 4 |
| 7 | Session Protocol Type | ACME | 43 |
| 8 | Session-Forked-Call-Id | ACME | 171 |
| 9 | Generic ID | ACME | 40 |
| 10 | Calling Station ID | none | 31 |
| 11 | Called Station ID | none | 30 |
| 12 | Cisco Setup Time | CISCO | 25 |
| 13 | Cisco Connect Time | CISCO | 28 |
| 14 | Egress Network Interface ID | ACME | 139 |
| 15 | Egress Vlan Tag Value | ACME | 140 |
| 16 | Ingress Network Interface ID | ACME | 137 |
| 17 | Ingress Vlan Tag Value | ACME | 138 |
| 18 | Egress Realm | ACME | 42 |
| 19 | Ingress Realm | ACME | 41 |
| 20 | Flow Identifier | ACME | 1 |
| 21 | Flow Type | ACME | 2 |
| 22 | Flow Input Realm | ACME | 10 |

Table B-1 (Cont.) Local CDR Start Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|-------------------------|--------|------------|
| 23 | Flow Input Src Addr | ACME | 11 |
| 24 | Flow Input Src Port | ACME | 12 |
| 25 | Flow Input Dest Address | ACME | 13 |
| 26 | Flow Input Dest Port | ACME | 14 |
| 27 | Flow Output Realm | ACME | 20 |
| 28 | Flow Output Src Address | ACME | 21 |
| 29 | Flow Output Src Port | ACME | 22 |
| 30 | Flow Output Dest Addr | ACME | 23 |
| 31 | Flow Output Dest Port | ACME | 24 |
| 32 | Flow Identifier | ACME | 78 |
| 33 | Flow Type | ACME | 79 |
| 34 | Flow Input Realm | ACME | 80 |
| 35 | Flow Input Src Addr | ACME | 81 |
| 36 | Flow Input Src Port | ACME | 82 |
| 37 | Flow Input Dest Address | ACME | 83 |
| 38 | Flow Input Dest Port | ACME | 84 |
| 39 | Flow Output Realm | ACME | 85 |
| 40 | Flow Output Src Address | ACME | 86 |
| 41 | Flow Output Src Port | ACME | 87 |
| 42 | Flow Output Dest Addr | ACME | 88 |
| 43 | Flow Output Dest Port | ACME | 89 |
| 44 | Flow Identifier | ACME | 90 |
| 45 | Flow Type | ACME | 91 |
| 46 | Flow Input Realm | ACME | 92 |
| 47 | Flow Input Src Addr | ACME | 93 |
| 48 | Flow Input Src Port | ACME | 94 |
| 49 | Flow Input Dest Address | ACME | 95 |
| 50 | Flow Input Dest Port | ACME | 96 |
| 51 | Flow Output Realm | ACME | 97 |
| 52 | Flow Output Src Address | ACME | 98 |
| 53 | Flow Output Src Port | ACME | 99 |
| 54 | Flow Output Dest Addr | ACME | 100 |
| 55 | Flow Output Dest Port | ACME | 101 |
| 56 | Flow Identifier | ACME | 112 |
| 57 | Flow Type | ACME | 113 |
| 58 | Flow Input Realm | ACME | 114 |
| 59 | Flow Input Src Addr | ACME | 115 |
| 60 | Flow Input Src Port | ACME | 116 |
| 61 | Flow Input Dest Address | ACME | 117 |
| 62 | Flow Input Dest Port | ACME | 118 |
| 63 | Flow Output Realm | ACME | 119 |
| 64 | Flow Output Src Address | ACME | 120 |
| 65 | Flow Output Src Port | ACME | 121 |
| 66 | Flow Output Dest Addr | ACME | 122 |

Table B-1 (Cont.) Local CDR Start Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------|--------|------------|
| 67 | Flow Output Dest Port | ACME | 123 |
| 68 | Charging Vector ICID | ACME | 54 |
| 69 | Charging Function Address | ACME | 55 |
| 70 | Firmware Version | ACME | 56 |
| 71 | Local timezone | ACME | 57 |
| 72 | Post Dial Delay (msec) | ACME | 58 |
| 73 | Primary routing Number | ACME | 64 |
| 74 | Originating Trunk Group | ACME | 65 |
| 75 | Terminating Trunk Group | ACME | 66 |
| 76 | Originating Trunk Context | ACME | 67 |
| 77 | Terminating Trunk Context | ACME | 68 |
| 78 | P Asserted ID | ACME | 69 |
| 79 | Ingress Local Address | ACME | 74 |
| 80 | Ingress Remote Address | ACME | 75 |
| 81 | Egress Local Address | ACME | 76 |
| 82 | Egress Remote Address | ACME | 77 |
| 83 | SIP DIVERSION | ACME | 70 |
| 84 | Egress Routing Number | ACME | 134 |
| 85 | Ingress RPH | ACME | 135 |
| 86 | Egress RPH | ACME | 136 |
| 87 | Custom VSA 200 | ACME | 200 |
| 88 | Custom VSA 201 | ACME | 201 |
| 89 | Custom VSA 202 | ACME | 202 |
| 90 | Custom VSA 203 | ACME | 203 |
| 91 | Custom VSA 204 | ACME | 204 |
| 92 | Custom VSA 205 | ACME | 205 |
| 93 | Custom VSA 206 | ACME | 206 |
| 94 | Custom VSA 207 | ACME | 207 |
| 95 | Custom VSA 208 | ACME | 208 |
| 96 | Custom VSA 209 | ACME | 209 |
| 97 | Custom VSA 210 | ACME | 210 |
| 98 | Custom VSA 211 | ACME | 211 |
| 99 | Custom VSA 212 | ACME | 212 |
| 100 | Custom VSA 213 | ACME | 213 |
| 101 | Custom VSA 214 | ACME | 214 |
| 102 | Custom VSA 215 | ACME | 215 |
| 103 | Custom VSA 216 | ACME | 216 |
| 104 | Custom VSA 217 | ACME | 217 |
| 105 | Custom VSA 218 | ACME | 218 |
| 106 | Custom VSA 219 | ACME | 219 |
| 107 | Custom VSA 220 | ACME | 220 |
| 108 | Custom VSA 221 | ACME | 221 |
| 109 | Custom VSA 222 | ACME | 222 |
| 110 | Custom VSA 223 | ACME | 223 |

Table B-1 (Cont.) Local CDR Start Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------------------|--------|------------|
| 111 | Custom VSA 224 | ACME | 224 |
| 112 | Custom VSA 225 | ACME | 225 |
| 113 | Custom VSA 226 | ACME | 226 |
| 114 | Custom VSA 227 | ACME | 227 |
| 115 | Custom VSA 228 | ACME | 228 |
| 116 | Custom VSA 229 | ACME | 229 |
| 117 | Custom VSA 230 | ACME | 230 |
| 118 | Calling-Media-Stop-Time | ACME | 231 |
| 119 | Called-Media-Stop-Time | ACME | 232 |
| 120 | Calling-Media-Stop-Time | ACME | 233 |
| 121 | Called-Media-Stop-Time | ACME | 234 |
| 122 | Flow Media Type | ACME | 142 |
| 123 | Flow Media Type | ACME | 143 |
| 124 | Flow Media Type | ACME | 144 |
| 125 | Flow Media Type | ACME | 145 |
| 126 | PGW-IP Address | ACME | 249 |
| 127 | SGW-IP Address | ACME | 249 |
| 128 | ORIG-IOI | ACME | 249 |
| 129 | TERM-IOI | ACME | 249 |
| 130 | IMEI | ACME | 249 |
| 131 | Node Functionality | ACME | 249 |
| 132 | History Info | ACME | 250 |
| 133 | P-Visited Network ID | ACME | 251 |
| 134 | IMSI | ACME | 252 |
| 135 | Access-Network-Information | ACME | 248 |
| 136 | CDR Sequence Number | ACME | 59 |
| 137 | Stir-Signed-Request | ACME | 249 |
| 138 | Stir-Signed-Request-Exception-Id | ACME | 249 |
| 139 | Stir-Verified-Request | ACME | 249 |
| 140 | Stir-Verified-Request-Exception-Id | ACME | 249 |
| 141 | Stir-VS-Verstat | ACME | 249 |
| 142 | Stir-VS-Reason | ACME | 249 |
| 143 | Stir-TN-Used-For-AS-VS-Request | ACME | 249 |
| 144 | Stir-Div-Signed-Request | ACME | 249 |
| 145 | Stir-Div-Verified-Request | ACME | 249 |
| 146 | History Info2 | ACME | 253 |
| 147 | Stir-VS-Invite-State | ACME | 249 |
| 148 | Acme-SIP-HDR-TO | ACME | 249 |
| 149 | Stir-VS-attest | ACME | 178 |
| 150 | Stir-Div-VS-Reason | ACME | 179 |
| 151 | Stir-Div-VS-Verstat | ACME | 180 |
| 152 | Stir-AS-Bypass-Token | ACME | 181 |

Local CDR Interim Record (RADIUS)

The following table contains the local CDR layout, inclusive of all VSA data for an interim record, when RADIUS is selected as the account protocol.

Table B-2 Local CDR Interim Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 1 | Accounting Status | none | 40 |
| 2 | NAS IP Address | none | 4 |
| 3 | NAS Port | none | 5 |
| 4 | Accounting Session ID | none | 44 |
| 5 | Ingress Session ID | ACME | 3 |
| 6 | Egress Session ID | ACME | 4 |
| 7 | Session Protocol Type | ACME | 43 |
| 8 | Session-Forked-Call-Id | ACME | 171 |
| 9 | Generic ID | ACME | 40 |
| 10 | Calling Station ID | none | 31 |
| 11 | Called Station ID | none | 30 |
| 12 | Cisco Setup Time | CISCO | 25 |
| 13 | Cisco Connect Time | CISCO | 28 |
| 14 | Egress Network Interface ID | ACME | 139 |
| 15 | Egress Vlan Tag Value | ACME | 140 |
| 16 | Ingress Network Interface ID | ACME | 137 |
| 17 | Ingress Vlan Tag Value | ACME | 138 |
| 18 | Egress Realm | ACME | 42 |
| 19 | Ingress Realm | ACME | 41 |
| 20 | Flow Identifier | ACME | 1 |
| 21 | Flow Type | ACME | 2 |
| 22 | Flow Input Realm | ACME | 10 |
| 23 | Flow Input Src Addr | ACME | 11 |
| 24 | Flow Input Src Port | ACME | 12 |
| 25 | Flow Input Dest Address | ACME | 13 |
| 26 | Flow Input Dest Port | ACME | 14 |
| 27 | Flow Output Realm | ACME | 20 |
| 28 | Flow Output Src Address | ACME | 21 |
| 29 | Flow Output Src Port | ACME | 22 |
| 30 | Flow Output Dest Addr | ACME | 23 |
| 31 | Flow Output Dest Port | ACME | 24 |
| 32 | RTCP Calling Packets Lost | ACME | 32 |
| 33 | RTCP Calling Avg Jitter (msec) | ACME | 33 |
| 34 | RTCP Calling Avg Latency (msec) | ACME | 34 |
| 35 | RTCP Calling Max Jitter (msec) | ACME | 35 |
| 36 | RTCP Calling Max Latency (msec) | ACME | 36 |
| 37 | RTP Calling Packets Lost | ACME | 37 |
| 38 | RTP Calling Avg Jitter (msec) | ACME | 38 |
| 39 | RTP Calling Max Jitter (msec) | ACME | 39 |

Table B-2 (Cont.) Local CDR Interim Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 40 | RTP Calling Octets | ACME | 28 |
| 41 | RTP Calling Packets | ACME | 29 |
| 42 | Calling R-Factor | ACME | 151 |
| 43 | Calling MOS | ACME | 152 |
| 44 | Flow Identifier | ACME | 78 |
| 45 | Flow Type | ACME | 79 |
| 46 | Flow Input Realm | ACME | 80 |
| 47 | Flow Input Src Addr | ACME | 81 |
| 48 | Flow Input Src Port | ACME | 82 |
| 49 | Flow Input Dest Address | ACME | 83 |
| 50 | Flow Input Dest Port | ACME | 84 |
| 51 | Flow Output Realm | ACME | 85 |
| 52 | Flow Output Src Address | ACME | 86 |
| 53 | Flow Output Src Port | ACME | 87 |
| 54 | Flow Output Dest Addr | ACME | 88 |
| 55 | Flow Output Dest Port | ACME | 89 |
| 56 | RTCP Called Packets Lost | ACME | 46 |
| 57 | RTCP Called Avg Jitter (msec) | ACME | 47 |
| 58 | RTCP Called Avg Latency (msec) | ACME | 48 |
| 59 | RTCP Called Max Jitter (msec) | ACME | 49 |
| 60 | RTCP Called Max Latency (msec) | ACME | 50 |
| 61 | RTP Called Packets Lost | ACME | 51 |
| 62 | RTP Called Avg Jitter (msec) | ACME | 52 |
| 63 | RTP Called Max Jitter (msec) | ACME | 53 |
| 64 | RTP Called Octets | ACME | 44 |
| 65 | RTP Called Packets | ACME | 45 |
| 66 | Called R-Factor | ACME | 153 |
| 67 | Called MOS | ACME | 154 |
| 68 | Flow Identifier | ACME | 90 |
| 69 | Flow Type | ACME | 91 |
| 70 | Flow Input Realm | ACME | 92 |
| 71 | Flow Input Src Addr | ACME | 93 |
| 72 | Flow Input Src Port | ACME | 94 |
| 73 | Flow Input Dest Address | ACME | 95 |
| 74 | Flow Input Dest Port | ACME | 96 |
| 75 | Flow Output Realm | ACME | 97 |
| 76 | Flow Output Src Address | ACME | 98 |
| 77 | Flow Output Src Port | ACME | 99 |
| 78 | Flow Output Dest Addr | ACME | 100 |
| 79 | Flow Output Dest Port | ACME | 101 |
| 80 | RTCP Calling Packets Lost | ACME | 104 |
| 81 | RTCP Calling Avg Jitter (msec) | ACME | 105 |
| 82 | RTCP Calling Avg Latency (msec) | ACME | 106 |
| 83 | RTCP Calling Max Jitter (msec) | ACME | 107 |

Table B-2 (Cont.) Local CDR Interim Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 84 | RTCP Calling Max Latency (msec) | ACME | 108 |
| 85 | RTP Calling Packets Lost | ACME | 109 |
| 86 | RTP Calling Avg Jitter (msec) | ACME | 110 |
| 87 | RTP Calling Max Jitter (msec) | ACME | 111 |
| 88 | RTP Calling Octets | ACME | 102 |
| 89 | RTP Calling Packets | ACME | 103 |
| 90 | Flow Identifier | ACME | 112 |
| 91 | Flow Type | ACME | 113 |
| 92 | Flow Input Realm | ACME | 114 |
| 93 | Flow Input Src Addr | ACME | 115 |
| 94 | Flow Input Src Port | ACME | 116 |
| 95 | Flow Input Dest Address | ACME | 117 |
| 96 | Flow Input Dest Port | ACME | 118 |
| 97 | Flow Output Realm | ACME | 119 |
| 98 | Flow Output Src Address | ACME | 120 |
| 99 | Flow Output Src Port | ACME | 121 |
| 100 | Flow Output Dest Addr | ACME | 122 |
| 101 | Flow Output Dest Port | ACME | 123 |
| 102 | RTCP Called Packets Lost | ACME | 126 |
| 103 | RTCP Called Avg Jitter (msec) | ACME | 127 |
| 104 | RTCP Called Avg Latency (msec) | ACME | 128 |
| 105 | RTCP Called Max Jitter (msec) | ACME | 129 |
| 106 | RTCP Called Max Latency (msec) | ACME | 130 |
| 107 | RTP Called Packets Lost | ACME | 131 |
| 108 | RTP Called Avg Jitter (msec) | ACME | 132 |
| 109 | RTP Called Max Jitter (msec) | ACME | 133 |
| 110 | RTP Called Octets | ACME | 124 |
| 111 | RTP Called Packets | ACME | 125 |
| 112 | Charging Vector ICID | ACME | 54 |
| 113 | Charging Function Address | ACME | 55 |
| 114 | Firmware Version | ACME | 56 |
| 115 | Local timezone | ACME | 57 |
| 116 | Post Dial Delay (msec) | ACME | 58 |
| 117 | Primary routing Number | ACME | 64 |
| 118 | Originating Trunk Group | ACME | 65 |
| 119 | Terminating Trunk Group | ACME | 66 |
| 120 | Originating Trunk Context | ACME | 67 |
| 121 | Terminating Trunk Context | ACME | 68 |
| 122 | P Asserted ID | ACME | 69 |
| 123 | Ingress Local Address | ACME | 74 |
| 124 | Ingress Remote Address | ACME | 75 |
| 125 | Egress Local Address | ACME | 76 |
| 126 | Egress Remote Address | ACME | 77 |
| 127 | SIP DIVERSION | ACME | 70 |

Table B-2 (Cont.) Local CDR Interim Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|-------------------------|--------|------------|
| 128 | Intermediate Time | ACME | 63 |
| 129 | Accounting Session Time | NA | 46 |
| 130 | Egress Routing Number | ACME | 134 |
| 131 | Ingress RPH | ACME | 135 |
| 132 | Egress RPH | ACME | 136 |
| 133 | Custom VSA 200 | ACME | 200 |
| 134 | Custom VSA 201 | ACME | 201 |
| 135 | Custom VSA 202 | ACME | 202 |
| 136 | Custom VSA 203 | ACME | 203 |
| 137 | Custom VSA 204 | ACME | 204 |
| 138 | Custom VSA 205 | ACME | 205 |
| 139 | Custom VSA 206 | ACME | 206 |
| 140 | Custom VSA 207 | ACME | 207 |
| 141 | Custom VSA 208 | ACME | 208 |
| 142 | Custom VSA 209 | ACME | 209 |
| 143 | Custom VSA 210 | ACME | 210 |
| 144 | Custom VSA 211 | ACME | 211 |
| 145 | Custom VSA 212 | ACME | 212 |
| 146 | Custom VSA 213 | ACME | 213 |
| 147 | Custom VSA 214 | ACME | 214 |
| 148 | Custom VSA 215 | ACME | 215 |
| 149 | Custom VSA 216 | ACME | 216 |
| 150 | Custom VSA 217 | ACME | 217 |
| 151 | Custom VSA 218 | ACME | 218 |
| 152 | Custom VSA 219 | ACME | 219 |
| 153 | Custom VSA 220 | ACME | 220 |
| 154 | Custom VSA 221 | ACME | 221 |
| 155 | Custom VSA 222 | ACME | 222 |
| 156 | Custom VSA 223 | ACME | 223 |
| 157 | Custom VSA 224 | ACME | 224 |
| 158 | Custom VSA 225 | ACME | 225 |
| 159 | Custom VSA 226 | ACME | 226 |
| 160 | Custom VSA 227 | ACME | 227 |
| 161 | Custom VSA 228 | ACME | 228 |
| 162 | Custom VSA 229 | ACME | 229 |
| 163 | Custom VSA 230 | ACME | 230 |
| 164 | Calling-Media-Stop-Time | ACME | 231 |
| 165 | Called-Media-Stop-Time | ACME | 232 |
| 166 | Calling-Media-Stop-Time | ACME | 233 |
| 167 | Called-Media-Stop-Time | ACME | 234 |
| 168 | Flow Media Type | ACME | 142 |
| 169 | Flow Media Type | ACME | 143 |
| 170 | Flow Media Type | ACME | 144 |
| 171 | Flow Media Type | ACME | 145 |

Table B-2 (Cont.) Local CDR Interim Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------------------|--------|------------|
| 172 | PGW-IP Address | ACME | 249 |
| 173 | SGW-IP Address | ACME | 249 |
| 174 | ORIG-IOI | ACME | 249 |
| 175 | TERM-IOI | ACME | 249 |
| 176 | IMEI | ACME | 249 |
| 177 | Node Functionality | ACME | 249 |
| 178 | History Info | ACME | 250 |
| 179 | P-Visited Network ID | ACME | 251 |
| 180 | IMSI | ACME | 252 |
| 181 | Access-Network-Information | ACME | 248 |
| 182 | CDR Sequence Number | ACME | 59 |
| 183 | Stir-Signed-Request | ACME | 249 |
| 184 | Stir-Signed-Request-Exception-Id | ACME | 249 |
| 185 | Stir-Verified-Request | ACME | 249 |
| 186 | Stir-Verified-Request-Exception-Id | ACME | 249 |
| 187 | Stir-VS-Verstat | ACME | 249 |
| 188 | Stir-VS-Reason | ACME | 249 |
| 189 | Stir-TN-Used-For-AS-VS-Request | ACME | 249 |
| 190 | Stir-Div-Signed-Request | ACME | 249 |
| 191 | Stir-Div-Verified-Request | ACME | 249 |
| 192 | History Info2 | ACME | 253 |
| 193 | Stir-VS-Invite-State | ACME | 249 |
| 194 | Acme-SIP-HDR-TO | ACME | 249 |
| 195 | Stir-VS-attest | ACME | 178 |
| 196 | Stir-Div-VS-Reason | ACME | 179 |
| 197 | Stir-Div-VS-Verstat | ACME | 180 |
| 198 | Stir-AS-Bypass-Token | ACME | 181 |
| 206 | VBD Tone Detected | ACME | 182 |

Local CDR Stop Record (RADIUS)

The following table contains the local CDR layout, inclusive of all VSA data for a Stop record, when RADIUS is selected as the account protocol.

Table B-3 Local CDR Stop Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|-----------------------|--------|------------|
| 1 | Accounting Status | none | 40 |
| 2 | NAS IP Address | none | 4 |
| 3 | NAS Port | none | 5 |
| 4 | Accounting Session ID | none | 44 |
| 5 | Ingress Session ID | ACME | 3 |
| 6 | Egress Session ID | ACME | 4 |
| 7 | Session Protocol Type | ACME | 43 |

Table B-3 (Cont.) Local CDR Stop Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 8 | Session-Forked-Call-Id | ACME | 171 |
| 9 | Generic ID | ACME | 40 |
| 10 | Calling Station ID | none | 31 |
| 11 | Called Station ID | none | 30 |
| 12 | Accounting Termination Cause | none | 49 |
| 13 | Accounting Session Time | none | 46 |
| 14 | Cisco Setup Time | CISCO | 25 |
| 15 | Cisco Connect Time | CISCO | 28 |
| 16 | Cisco Disconnect Time | CISCO | 29 |
| 17 | Cisco Disconnect Cause | CISCO | 30 |
| 18 | Egress Network Interface ID | ACME | 139 |
| 19 | Egress Vlan Tag Value | ACME | 140 |
| 20 | Ingress Network Interface ID | ACME | 137 |
| 21 | Ingress Vlan Tag Value | ACME | 138 |
| 22 | Egress Realm | ACME | 42 |
| 23 | Ingress Realm | ACME | 41 |
| 24 | Flow Identifier | ACME | 1 |
| 25 | Flow Type | ACME | 2 |
| 26 | Flow Input Realm | ACME | 10 |
| 27 | Flow Input Src Addr | ACME | 11 |
| 28 | Flow Input Src Port | ACME | 12 |
| 29 | Flow Input Dest Address | ACME | 13 |
| 30 | Flow Input Dest Port | ACME | 14 |
| 31 | Flow Output Realm | ACME | 20 |
| 32 | Flow Output Src Address | ACME | 21 |
| 33 | Flow Output Src Port | ACME | 22 |
| 34 | Flow Output Dest Addr | ACME | 23 |
| 35 | Flow Output Dest Port | ACME | 24 |
| 36 | RTCP Calling Packets Lost | ACME | 32 |
| 37 | RTCP Calling Avg Jitter (msec) | ACME | 33 |
| 38 | RTCP Calling Avg Latency (msec) | ACME | 34 |
| 39 | RTCP Calling Max Jitter (msec) | ACME | 35 |
| 40 | RTCP Calling Max Latency (msec) | ACME | 36 |
| 41 | RTP Calling Packets Lost | ACME | 37 |
| 42 | RTP Calling Avg Jitter (msec) | ACME | 38 |
| 43 | RTP Calling Max Jitter (msec) | ACME | 39 |
| 44 | RTP Calling Octets | ACME | 28 |
| 45 | RTP Calling Packets | ACME | 29 |
| 46 | Calling R-Factor | ACME | 151 |
| 47 | Calling MOS | ACME | 152 |
| 48 | Flow Identifier | ACME | 78 |
| 49 | Flow Type | ACME | 79 |
| 50 | Flow Input Realm | ACME | 80 |
| 51 | Flow Input Src Addr | ACME | 81 |

Table B-3 (Cont.) Local CDR Stop Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 52 | Flow Input Src Port | ACME | 82 |
| 53 | Flow Input Dest Address | ACME | 83 |
| 54 | Flow Input Dest Port | ACME | 84 |
| 55 | Flow Output Realm | ACME | 85 |
| 56 | Flow Output Src Address | ACME | 86 |
| 57 | Flow Output Src Port | ACME | 87 |
| 58 | Flow Output Dest Addr | ACME | 88 |
| 59 | Flow Output Dest Port | ACME | 89 |
| 60 | RTCP Called Packets Lost | ACME | 46 |
| 61 | RTCP Called Avg Jitter (msec) | ACME | 47 |
| 62 | RTCP Called Avg Latency (msec) | ACME | 48 |
| 63 | RTCP Called Max Jitter (msec) | ACME | 49 |
| 64 | RTCP Called Max Latency (msec) | ACME | 50 |
| 65 | RTP Called Packets Lost | ACME | 51 |
| 66 | RTP Called Avg Jitter (msec) | ACME | 52 |
| 67 | RTP Called Max Jitter (msec) | ACME | 53 |
| 68 | RTP Called Octets | ACME | 44 |
| 69 | RTP Called Packets | ACME | 45 |
| 70 | Called R-Factor | ACME | 153 |
| 71 | Called MOS | ACME | 154 |
| 72 | Flow Identifier | ACME | 90 |
| 73 | Flow Type | ACME | 91 |
| 74 | Flow Input Realm | ACME | 92 |
| 75 | Flow Input Src Addr | ACME | 93 |
| 76 | Flow Input Src Port | ACME | 94 |
| 77 | Flow Input Dest Address | ACME | 95 |
| 78 | Flow Input Dest Port | ACME | 96 |
| 79 | Flow Output Realm | ACME | 97 |
| 80 | Flow Output Src Address | ACME | 98 |
| 81 | Flow Output Src Port | ACME | 99 |
| 82 | Flow Output Dest Addr | ACME | 100 |
| 83 | Flow Output Dest Port | ACME | 101 |
| 84 | RTCP Calling Packets Lost | ACME | 104 |
| 85 | RTCP Calling Avg Jitter (msec) | ACME | 105 |
| 86 | RTCP Calling Avg Latency (msec) | ACME | 106 |
| 87 | RTCP Calling Max Jitter (msec) | ACME | 107 |
| 88 | RTCP Calling Max Latency (msec) | ACME | 108 |
| 89 | RTP Calling Packets Lost | ACME | 109 |
| 90 | RTP Calling Avg Jitter (msec) | ACME | 110 |
| 91 | RTP Calling Max Jitter (msec) | ACME | 111 |
| 92 | RTP Calling Octets | ACME | 102 |
| 93 | RTP Calling Packets | ACME | 103 |
| 94 | Flow Identifier | ACME | 112 |
| 95 | Flow Type | ACME | 113 |

Table B-3 (Cont.) Local CDR Stop Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|--------------------------------|--------|------------|
| 96 | Flow Input Realm | ACME | 114 |
| 97 | Flow Input Src Addr | ACME | 115 |
| 98 | Flow Input Src Port | ACME | 116 |
| 99 | Flow Input Dest Address | ACME | 117 |
| 100 | Flow Input Dest Port | ACME | 118 |
| 101 | Flow Output Realm | ACME | 119 |
| 102 | Flow Output Src Address | ACME | 120 |
| 103 | Flow Output Src Port | ACME | 121 |
| 104 | Flow Output Dest Addr | ACME | 122 |
| 105 | Flow Output Dest Port | ACME | 123 |
| 106 | RTCP Called Packets Lost | ACME | 126 |
| 107 | RTCP Called Avg Jitter (msec) | ACME | 127 |
| 108 | RTCP Called Avg Latency (msec) | ACME | 128 |
| 109 | RTCP Called Max Jitter (msec) | ACME | 129 |
| 110 | RTCP Called Max Latency (msec) | ACME | 130 |
| 111 | RTP Called Packets Lost | ACME | 131 |
| 112 | RTP Called Avg Jitter (msec) | ACME | 132 |
| 113 | RTP Called Max Jitter (msec) | ACME | 133 |
| 114 | RTP Called Octets | ACME | 124 |
| 115 | RTP Called Packets | ACME | 125 |
| 116 | Charging Vector ICID | ACME | 54 |
| 117 | Charging Function Address | ACME | 55 |
| 118 | Firmware Version | ACME | 56 |
| 119 | Local timezone | ACME | 57 |
| 120 | Post Dial Delay (msec) | ACME | 58 |
| 121 | Primary routing Number | ACME | 64 |
| 122 | Originating Trunk Group | ACME | 65 |
| 123 | Terminating Trunk Group | ACME | 66 |
| 124 | Originating Trunk Context | ACME | 67 |
| 125 | Terminating Trunk Context | ACME | 68 |
| 126 | P Asserted ID | ACME | 69 |
| 127 | Ingress Local Address | ACME | 74 |
| 128 | Ingress Remote Address | ACME | 75 |
| 129 | Egress Local Address | ACME | 76 |
| 130 | Egress Remote Address | ACME | 77 |
| 131 | SIP DIVERSION | ACME | 70 |
| 132 | Session Disposition | ACME | 60 |
| 133 | Disconnect Initiator | ACME | 61 |
| 134 | Disconnect Cause | ACME | 62 |
| 135 | Sip Status Code | ACME | 71 |
| 136 | Egress Routing Number | ACME | 134 |
| 137 | Ingress RPH | ACME | 135 |
| 138 | Egress RPH | ACME | 136 |
| 139 | refer call transfer ID | ACME | 141 |

Table B-3 (Cont.) Local CDR Stop Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 140 | Custom VSA 200 | ACME | 200 |
| 141 | Custom VSA 201 | ACME | 201 |
| 142 | Custom VSA 202 | ACME | 202 |
| 143 | Custom VSA 203 | ACME | 203 |
| 144 | Custom VSA 204 | ACME | 204 |
| 145 | Custom VSA 205 | ACME | 205 |
| 146 | Custom VSA 206 | ACME | 206 |
| 147 | Custom VSA 207 | ACME | 207 |
| 148 | Custom VSA 208 | ACME | 208 |
| 149 | Custom VSA 209 | ACME | 209 |
| 150 | Custom VSA 210 | ACME | 210 |
| 151 | Custom VSA 211 | ACME | 211 |
| 152 | Custom VSA 212 | ACME | 212 |
| 153 | Custom VSA 213 | ACME | 213 |
| 154 | Custom VSA 214 | ACME | 214 |
| 155 | Custom VSA 215 | ACME | 215 |
| 156 | Custom VSA 216 | ACME | 216 |
| 157 | Custom VSA 217 | ACME | 217 |
| 158 | Custom VSA 218 | ACME | 218 |
| 159 | Custom VSA 219 | ACME | 219 |
| 160 | Custom VSA 220 | ACME | 220 |
| 161 | Custom VSA 221 | ACME | 221 |
| 162 | Custom VSA 222 | ACME | 222 |
| 163 | Custom VSA 223 | ACME | 223 |
| 164 | Custom VSA 224 | ACME | 224 |
| 165 | Custom VSA 225 | ACME | 225 |
| 166 | Custom VSA 226 | ACME | 226 |
| 167 | Custom VSA 227 | ACME | 227 |
| 168 | Custom VSA 228 | ACME | 228 |
| 169 | Custom VSA 229 | ACME | 229 |
| 170 | Custom VSA 230 | ACME | 230 |
| 171 | Calling-Media-Stop-Time | ACME | 231 |
| 172 | Called-Media-Stop-Time | ACME | 232 |
| 173 | Calling-Media-Stop-Time | ACME | 233 |
| 174 | Called-Media-Stop-Time | ACME | 234 |
| 175 | Flow Media Type | ACME | 142 |
| 176 | Flow Media Type | ACME | 143 |
| 177 | Flow Media Type | ACME | 144 |
| 178 | Flow Media Type | ACME | 145 |
| 179 | RTP Calling Octets Transmitted | ACME | 240 |
| 180 | RTP Calling Packets Transmitted | ACME | 241 |
| 181 | RTP Called Octets Transmitted | ACME | 242 |
| 182 | RTP Called Packets Transmitted | ACME | 243 |
| 183 | RTP Calling Octets Transmitted | ACME | 244 |

Table B-3 (Cont.) Local CDR Stop Record (RADIUS)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------------------|--------|------------|
| 184 | RTP Calling Packets Transmitted | ACME | 245 |
| 185 | RTP Called Octets Transmitted | ACME | 244 |
| 186 | RTP Called Packets Transmitted | ACME | 245 |
| 187 | MSRP Called Octets | ACME | 249 |
| 188 | MSRP Called Packets | ACME | 249 |
| 189 | MSRP Called Octets Transmitted | ACME | 249 |
| 190 | MSRP Called Packets Transmitted | ACME | 249 |
| 191 | MSRP Calling Octets | ACME | 249 |
| 192 | MSRP Calling Packets | ACME | 249 |
| 193 | MSRP Calling Octets Transmitted | ACME | 249 |
| 194 | MSRP Calling Packets Transmitted | ACME | 249 |
| 195 | PGW-IP Address | ACME | 249 |
| 196 | SGW-IP Address | ACME | 249 |
| 197 | ORIG-IOI | ACME | 249 |
| 198 | TERM-IOI | ACME | 249 |
| 199 | IMEI | ACME | 249 |
| 200 | Node Functionality | ACME | 249 |
| 201 | History Info | ACME | 250 |
| 202 | P-Visited Network ID | ACME | 251 |
| 203 | IMSI | ACME | 252 |
| 204 | Access-Network-Information | ACME | 248 |
| 205 | CDR Sequence Number | ACME | 59 |
| 206 | Stir-Signed-Request | ACME | 249 |
| 207 | Stir-Signed-Request-Exception-Id | ACME | 249 |
| 208 | Stir-Verified-Request | ACME | 249 |
| 209 | Stir-Verified-Request-Exception-Id | ACME | 249 |
| 210 | Stir-VS-Verstat | ACME | 249 |
| 211 | Stir-VS-Reason | ACME | 249 |
| 212 | Stir-TN-Used-For-AS-VS-Request | ACME | 249 |
| 213 | Stir-Div-Signed-Request | ACME | 249 |
| 214 | Stir-Div-Verified-Request | ACME | 249 |
| 215 | History Info2 | ACME | 253 |
| 216 | Stir-VS-Invite-State | ACME | 249 |
| 217 | Acme-SIP-HDR-TO | ACME | 249 |
| 218 | Stir-VS-attest | ACME | 178 |
| 219 | Stir-Div-VS-Reason | ACME | 179 |
| 220 | Stir-Div-VS-Verstat | ACME | 180 |
| 221 | Stir-AS-Bypass-Token | ACME | 181 |
| 229 | VBD Tone Detected | ACME | 182 |

Local CDR Message Record for SMS (RADIUS)

The SBC creates an Message CSV file for SMS messages when the generate-event = message parameter is enabled. The following table shows the inclusive CSV element order, when protocol = RADIUS (or protocol=default).

Table B-4 Local CDR Message Record (RADIUS)

| CSV Position | Attribute Name | Vendor ID | CDR Number |
|--------------|---------------------------------|-----------|------------|
| 1 | "Accounting Status" | none | 40 |
| 2 | "NAS IP Address" | none | 4 |
| 3 | "NAS Port" | none | 5 |
| 4 | "Accounting Session ID" | none | 44 |
| 5 | "Ingress Session ID" | ACME | 3 |
| 6 | "Egress Session ID" | ACME | 4 |
| 7 | "Session Protocol Type" | ACME | 43 |
| 8 | "Session-Forked-Call-Id" | ACME | 171 |
| 9 | "Generic ID" | ACME | 40 |
| 10 | "Calling Station ID" | none | 31 |
| 11 | "Called Station ID" | none | 30 |
| 12 | "Egress Network Interface ID " | ACME | 139 |
| 13 | "Egress Vlan Tag Value " | ACME | 140 |
| 14 | "Ingress Network Interface ID " | ACME | 137 |
| 15 | "Ingress Vlan Tag Value " | ACME | 138 |
| 16 | "Cisco Setup Time" | CISCO | 25 |
| 17 | "Cisco Connect Time" | CISCO | 28 |
| 18 | "Custom VSA 200" | ACME | 200 |
| 19 | "Custom VSA 201" | ACME | 201 |
| 20 | "Custom VSA 202" | ACME | 202 |
| 21 | "Custom VSA 203" | ACME | 203 |
| 22 | "Custom VSA 204" | ACME | 204 |
| 23 | "Custom VSA 205" | ACME | 205 |
| 24 | "Custom VSA 206" | ACME | 206 |
| 25 | "Custom VSA 207" | ACME | 207 |
| 26 | "Custom VSA 208" | ACME | 208 |
| 27 | "Custom VSA 209" | ACME | 209 |
| 28 | "Custom VSA 210" | ACME | 210 |
| 29 | "Custom VSA 211" | ACME | 211 |
| 30 | "Custom VSA 212" | ACME | 212 |
| 31 | "Custom VSA 213" | ACME | 213 |
| 32 | "Custom VSA 214" | ACME | 214 |
| 33 | "Custom VSA 215" | ACME | 215 |
| 34 | "Custom VSA 216" | ACME | 216 |
| 35 | "Custom VSA 217" | ACME | 217 |
| 36 | "Custom VSA 218" | ACME | 218 |
| 37 | "Custom VSA 219" | ACME | 219 |
| 38 | "Custom VSA 220" | ACME | 220 |

Table B-4 (Cont.) Local CDR Message Record (RADIUS)

| CSV Position | Attribute Name | Vendor ID | CDR Number |
|--------------|------------------------------|-----------|------------|
| 39 | "Custom VSA 221" | ACME | 221 |
| 40 | "Custom VSA 222" | ACME | 222 |
| 41 | "Custom VSA 223" | ACME | 223 |
| 42 | "Custom VSA 224" | ACME | 224 |
| 43 | "Custom VSA 225" | ACME | 225 |
| 44 | "Custom VSA 226" | ACME | 226 |
| 45 | "Custom VSA 227" | ACME | 227 |
| 46 | "Custom VSA 228" | ACME | 228 |
| 47 | "Custom VSA 229" | ACME | 229 |
| 48 | "Custom VSA 230" | ACME | 230 |
| 49 | "PGW-IP Address" | ACME | 249 |
| 50 | "SGW-IP Address" | ACME | 249 |
| 51 | "ORIG-IOI" | ACME | 249 |
| 52 | "TERM-IOI" | ACME | 249 |
| 53 | "IMEI" | ACME | 249 |
| 54 | "Node Functionality" | ACME | 249 |
| 55 | "Sms-Msg-Type" | ACME | 249 |
| 56 | "Sms-Calling-Party-Number" | ACME | 249 |
| 57 | "Sms-Called-Party-Number" | ACME | 249 |
| 58 | "Sms-Msg-length" | ACME | 249 |
| 59 | "P-Visited Network ID" | ACME | 251 |
| 60 | "IMSI" | ACME | 252 |
| 61 | "Access-Network-Information" | ACME | 248 |
| 62 | "CDR Sequence Number " | ACME | 59 |

Local CDR Start Record (Diameter)

The following table contains the local CDR layout, inclusive of all VSA data for a start record, when Diameter is selected as the account protocol.

Table B-5 Local CDR Start Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------|--------|------------|
| 1 | Accounting Status | none | 40 |
| 2 | NAS IP Address | none | 4 |
| 3 | NAS Port | none | 5 |
| 4 | Accounting Session ID | none | 44 |
| 5 | Ingress Session ID | ACME | 3 |
| 6 | Egress Session ID | ACME | 4 |
| 7 | Session Protocol Type | ACME | 43 |
| 8 | Session-Forked-Call-Id | ACME | 171 |
| 9 | Generic ID | ACME | 40 |
| 10 | Calling Station ID | none | 31 |
| 11 | Called Station ID | none | 30 |

Table B-5 (Cont.) Local CDR Start Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------------|--------|------------|
| 12 | Cisco Setup Time | CISCO | 25 |
| 13 | Cisco Connect Time | CISCO | 28 |
| 14 | Egress Network Interface ID | ACME | 139 |
| 15 | Egress Vlan Tag Value | ACME | 140 |
| 16 | Ingress Network Interface ID | ACME | 137 |
| 17 | Ingress Vlan Tag Value | ACME | 138 |
| 18 | Egress Realm | ACME | 42 |
| 19 | Ingress Realm | ACME | 41 |
| 20 | Flow Identifier | ACME | 1 |
| 21 | Flow Type | ACME | 2 |
| 22 | Flow Input Realm | ACME | 10 |
| 23 | Flow Input Src Addr | ACME | 11 |
| 24 | Flow Input Src Port | ACME | 12 |
| 25 | Flow Input Dest Address | ACME | 13 |
| 26 | Flow Input Dest Port | ACME | 14 |
| 27 | Flow Output Realm | ACME | 20 |
| 28 | Flow Output Src Address | ACME | 21 |
| 29 | Flow Output Src Port | ACME | 22 |
| 30 | Flow Output Dest Addr | ACME | 23 |
| 31 | Flow Output Dest Port | ACME | 24 |
| 32 | Flow Identifier | ACME | 78 |
| 33 | Flow Type | ACME | 79 |
| 34 | Flow Input Realm | ACME | 80 |
| 35 | Flow Input Src Addr | ACME | 81 |
| 36 | Flow Input Src Port | ACME | 82 |
| 37 | Flow Input Dest Address | ACME | 83 |
| 38 | Flow Input Dest Port | ACME | 84 |
| 39 | Flow Output Realm | ACME | 85 |
| 40 | Flow Output Src Address | ACME | 86 |
| 41 | Flow Output Src Port | ACME | 87 |
| 42 | Flow Output Dest Addr | ACME | 88 |
| 43 | Flow Output Dest Port | ACME | 89 |
| 44 | Flow Identifier | ACME | 90 |
| 45 | Flow Type | ACME | 91 |
| 46 | Flow Input Realm | ACME | 92 |
| 47 | Flow Input Src Addr | ACME | 93 |
| 48 | Flow Input Src Port | ACME | 94 |
| 49 | Flow Input Dest Address | ACME | 95 |
| 50 | Flow Input Dest Port | ACME | 96 |
| 51 | Flow Output Realm | ACME | 97 |
| 52 | Flow Output Src Address | ACME | 98 |
| 53 | Flow Output Src Port | ACME | 99 |
| 54 | Flow Output Dest Addr | ACME | 100 |
| 55 | Flow Output Dest Port | ACME | 101 |

Table B-5 (Cont.) Local CDR Start Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------|--------|------------|
| 56 | Flow Identifier | ACME | 112 |
| 57 | Flow Type | ACME | 113 |
| 58 | Flow Input Realm | ACME | 114 |
| 59 | Flow Input Src Addr | ACME | 115 |
| 60 | Flow Input Src Port | ACME | 116 |
| 61 | Flow Input Dest Address | ACME | 117 |
| 62 | Flow Input Dest Port | ACME | 118 |
| 63 | Flow Output Realm | ACME | 119 |
| 64 | Flow Output Src Address | ACME | 120 |
| 65 | Flow Output Src Port | ACME | 121 |
| 66 | Flow Output Dest Addr | ACME | 122 |
| 67 | Flow Output Dest Port | ACME | 123 |
| 68 | Charging Vector ICID | ACME | 54 |
| 69 | Charging Function Address | ACME | 55 |
| 70 | Firmware Version | ACME | 56 |
| 71 | Local timezone | ACME | 57 |
| 72 | Post Dial Delay (msec) | ACME | 58 |
| 73 | Primary routing Number | ACME | 64 |
| 74 | Originating Trunk Group | ACME | 65 |
| 75 | Terminating Trunk Group | ACME | 66 |
| 76 | Originating Trunk Context | ACME | 67 |
| 77 | Terminating Trunk Context | ACME | 68 |
| 78 | P Asserted ID | ACME | 69 |
| 79 | Ingress Local Address | ACME | 74 |
| 80 | Ingress Remote Address | ACME | 75 |
| 81 | Egress Local Address | ACME | 76 |
| 82 | Egress Remote Address | ACME | 77 |
| 83 | SIP DIVERSION | ACME | 70 |
| 84 | Egress Routing Number | ACME | 134 |
| 85 | Ingress RPH | ACME | 135 |
| 86 | Egress RPH | ACME | 136 |
| 87 | Diam-Session-Id | ACME | 172 |
| 88 | Sip-Method | ACME | 173 |
| 89 | Event-Time | ACME | 176 |
| 90 | Content-Disposition | ACME | 4119 |
| 91 | Originator | ACME | 4120 |
| 92 | Subscription-ID-Data | ACME | 4110 |
| 93 | Subscription-ID-Type | ACME | 4111 |
| 94 | Service-Context-ID | ACME | 4121 |
| 95 | Called-Asserted-Identity | ACME | 4117 |
| 96 | SDP-Media | ACME | 177 |
| 97 | SDP-Type | ACME | 4114 |
| 98 | SDP-Offer-Timestamp | ACME | 4115 |
| 99 | SDP-Answer-Timestamp | ACME | 4116 |

Table B-5 (Cont.) Local CDR Start Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 100 | Early-SDP-Media | ACME | 4123 |
| 101 | Early-SDP-Type | ACME | 4122 |
| 102 | Served-Party-IP | ACME | 4098 |
| 103 | User-Name | ACME | 4097 |
| 104 | Node-Func | ACME | 4100 |
| 105 | ApplicationId | ACME | 4101 |
| 106 | Role-Of-Node | ACME | 4102 |
| 107 | SIP-Request-Timestamp | ACME | 4124 |
| 108 | SIP-Request-Timestamp-Fraction | ACME | 4112 |
| 109 | SIP-Response-Timestamp | ACME | 4125 |
| 110 | SIP-Response-Timestamp-Fraction | ACME | 4113 |
| 111 | Access-Network-Information | ACME | 248 |
| 112 | Custom VSA 200 | ACME | 200 |
| 113 | Custom VSA 201 | ACME | 201 |
| 114 | Custom VSA 202 | ACME | 202 |
| 115 | Custom VSA 203 | ACME | 203 |
| 116 | Custom VSA 204 | ACME | 204 |
| 117 | Custom VSA 205 | ACME | 205 |
| 118 | Custom VSA 206 | ACME | 206 |
| 119 | Custom VSA 207 | ACME | 207 |
| 120 | Custom VSA 208 | ACME | 208 |
| 121 | Custom VSA 209 | ACME | 209 |
| 122 | Custom VSA 210 | ACME | 210 |
| 123 | Custom VSA 211 | ACME | 211 |
| 124 | Custom VSA 212 | ACME | 212 |
| 125 | Custom VSA 213 | ACME | 213 |
| 126 | Custom VSA 214 | ACME | 214 |
| 127 | Custom VSA 215 | ACME | 215 |
| 128 | Custom VSA 216 | ACME | 216 |
| 129 | Custom VSA 217 | ACME | 217 |
| 130 | Custom VSA 218 | ACME | 218 |
| 131 | Custom VSA 219 | ACME | 219 |
| 132 | Custom VSA 220 | ACME | 220 |
| 133 | Custom VSA 221 | ACME | 221 |
| 134 | Custom VSA 222 | ACME | 222 |
| 135 | Custom VSA 223 | ACME | 223 |
| 136 | Custom VSA 224 | ACME | 224 |
| 137 | Custom VSA 225 | ACME | 225 |
| 138 | Custom VSA 226 | ACME | 226 |
| 139 | Custom VSA 227 | ACME | 227 |
| 140 | Custom VSA 228 | ACME | 228 |
| 141 | Custom VSA 229 | ACME | 229 |
| 142 | Custom VSA 230 | ACME | 230 |
| 143 | Calling-Media-Stop-Time | ACME | 231 |

Table B-5 (Cont.) Local CDR Start Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|--|--------|------------|
| 144 | Called-Media-Stop-Time | ACME | 232 |
| 145 | Calling-Media-Stop-Time | ACME | 233 |
| 146 | Called-Media-Stop-Time | ACME | 234 |
| 147 | Flow Media Type | ACME | 142 |
| 148 | Flow Media Type | ACME | 143 |
| 149 | Flow Media Type | ACME | 144 |
| 150 | Flow Media Type | ACME | 145 |
| 151 | PGW-IP Address | ACME | 249 |
| 152 | SGW-IP Address | ACME | 249 |
| 153 | ORIG-IOI | ACME | 249 |
| 154 | TERM-IOI | ACME | 249 |
| 155 | IMEI | ACME | 249 |
| 156 | Node Functionality | ACME | 249 |
| 157 | History Info | ACME | 250 |
| 158 | P-Visited Network ID | ACME | 251 |
| 159 | IMSI | ACME | 252 |
| 160 | CDR Sequence Number | ACME | 59 |
| 161 | Origin-Realm | ACME | 4103 |
| 162 | Origin-Host | ACME | 4104 |
| 163 | Destination-Realm | ACME | 4105 |
| 164 | Destination-Host | ACME | 4106 |
| 165 | Stir-Signed-Request | ACME | 249 |
| 166 | Stir-Signed-Request-Exception-Id | ACME | 249 |
| 167 | Stir-Verified-Request | ACME | 249 |
| 168 | Stir-Verified-Request-Exception-Id | ACME | 249 |
| 169 | Stir-VS-Verstat | ACME | 249 |
| 170 | Stir-VS-Reason | ACME | 249 |
| 171 | Stir-TN-Used-For-AS-VS-Request | ACME | 249 |
| 172 | Stir-Div-Signed-Request | ACME | 249 |
| 173 | Stir-Div-Verified-Request | ACME | 249 |
| 174 | History Info2 | ACME | 253 |
| 175 | Stir-VS-Invite-State | ACME | 249 |
| 176 | Acme-SIP-HDR-TO | ACME | 249 |
| 177 | Stir-VS-attest | ACME | 178 |
| 178 | Stir-Div-VS-Reason | ACME | 179 |
| 179 | Stir-Div-VS-Verstat | ACME | 180 |
| 180 | Stir-AS-Bypass-Token | ACME | 181 |
| 181 | Number-Portability-Routing-Information | ACME | 2024 |

Local CDR Interim Record (Diameter)

The following table contains the local CDR layout, inclusive of all VSA data for a start record, when Diameter is selected as the account protocol.

Table B-6 Local CDR Interim Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 1 | Accounting Status | none | 40 |
| 2 | NAS IP Address | none | 4 |
| 3 | NAS Port | none | 5 |
| 4 | Accounting Session ID | none | 44 |
| 5 | Ingress Session ID | ACME | 3 |
| 6 | Egress Session ID | ACME | 4 |
| 7 | Session Protocol Type | ACME | 43 |
| 8 | Session-Forked-Call-Id | ACME | 171 |
| 9 | Generic ID | ACME | 40 |
| 10 | Calling Station ID | none | 31 |
| 11 | Called Station ID | none | 30 |
| 12 | Cisco Setup Time | CISCO | 25 |
| 13 | Cisco Connect Time | CISCO | 28 |
| 14 | Egress Network Interface ID | ACME | 139 |
| 15 | Egress Vlan Tag Value | ACME | 140 |
| 16 | Ingress Network Interface ID | ACME | 137 |
| 17 | Ingress Vlan Tag Value | ACME | 138 |
| 18 | Egress Realm | ACME | 42 |
| 19 | Ingress Realm | ACME | 41 |
| 20 | Flow Identifier | ACME | 1 |
| 21 | Flow Type | ACME | 2 |
| 22 | Flow Input Realm | ACME | 10 |
| 23 | Flow Input Src Addr | ACME | 11 |
| 24 | Flow Input Src Port | ACME | 12 |
| 25 | Flow Input Dest Address | ACME | 13 |
| 26 | Flow Input Dest Port | ACME | 14 |
| 27 | Flow Output Realm | ACME | 20 |
| 28 | Flow Output Src Address | ACME | 21 |
| 29 | Flow Output Src Port | ACME | 22 |
| 30 | Flow Output Dest Addr | ACME | 23 |
| 31 | Flow Output Dest Port | ACME | 24 |
| 32 | RTCP Calling Packets Lost | ACME | 32 |
| 33 | RTCP Calling Avg Jitter (msec) | ACME | 33 |
| 34 | RTCP Calling Avg Latency (msec) | ACME | 34 |
| 35 | RTCP Calling Max Jitter (msec) | ACME | 35 |
| 36 | RTCP Calling Max Latency (msec) | ACME | 36 |
| 37 | RTP Calling Packets Lost | ACME | 37 |
| 38 | RTP Calling Avg Jitter (msec) | ACME | 38 |
| 39 | RTP Calling Max Jitter (msec) | ACME | 39 |
| 40 | RTP Calling Octets | ACME | 28 |
| 41 | RTP Calling Packets | ACME | 29 |
| 42 | Calling R-Factor | ACME | 151 |
| 43 | Calling MOS | ACME | 152 |
| 44 | Flow Identifier | ACME | 78 |

Table B-6 (Cont.) Local CDR Interim Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 45 | Flow Type | ACME | 79 |
| 46 | Flow Input Realm | ACME | 80 |
| 47 | Flow Input Src Addr | ACME | 81 |
| 48 | Flow Input Src Port | ACME | 82 |
| 49 | Flow Input Dest Address | ACME | 83 |
| 50 | Flow Input Dest Port | ACME | 84 |
| 51 | Flow Output Realm | ACME | 85 |
| 52 | Flow Output Src Address | ACME | 86 |
| 53 | Flow Output Src Port | ACME | 87 |
| 54 | Flow Output Dest Addr | ACME | 88 |
| 55 | Flow Output Dest Port | ACME | 89 |
| 56 | RTCP Called Packets Lost | ACME | 46 |
| 57 | RTCP Called Avg Jitter (msec) | ACME | 47 |
| 58 | RTCP Called Avg Latency (msec) | ACME | 48 |
| 59 | RTCP Called Max Jitter (msec) | ACME | 49 |
| 60 | RTCP Called Max Latency (msec) | ACME | 50 |
| 61 | RTP Called Packets Lost | ACME | 51 |
| 62 | RTP Called Avg Jitter (msec) | ACME | 52 |
| 63 | RTP Called Max Jitter (msec) | ACME | 53 |
| 64 | RTP Called Octets | ACME | 44 |
| 65 | RTP Called Packets | ACME | 45 |
| 66 | Called R-Factor | ACME | 153 |
| 67 | Called MOS | ACME | 154 |
| 68 | Flow Identifier | ACME | 90 |
| 69 | Flow Type | ACME | 91 |
| 70 | Flow Input Realm | ACME | 92 |
| 71 | Flow Input Src Addr | ACME | 93 |
| 72 | Flow Input Src Port | ACME | 94 |
| 73 | Flow Input Dest Address | ACME | 95 |
| 74 | Flow Input Dest Port | ACME | 96 |
| 75 | Flow Output Realm | ACME | 97 |
| 76 | Flow Output Src Address | ACME | 98 |
| 77 | Flow Output Src Port | ACME | 99 |
| 78 | Flow Output Dest Addr | ACME | 100 |
| 79 | Flow Output Dest Port | ACME | 101 |
| 80 | RTCP Calling Packets Lost | ACME | 104 |
| 81 | RTCP Calling Avg Jitter (msec) | ACME | 105 |
| 82 | RTCP Calling Avg Latency (msec) | ACME | 106 |
| 83 | RTCP Calling Max Jitter (msec) | ACME | 107 |
| 84 | RTCP Calling Max Latency (msec) | ACME | 108 |
| 85 | RTP Calling Packets Lost | ACME | 109 |
| 86 | RTP Calling Avg Jitter (msec) | ACME | 110 |
| 87 | RTP Calling Max Jitter (msec) | ACME | 111 |
| 88 | RTP Calling Octets | ACME | 102 |

Table B-6 (Cont.) Local CDR Interim Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|--------------------------------|--------|------------|
| 89 | RTP Calling Packets | ACME | 103 |
| 90 | Flow Identifier | ACME | 112 |
| 91 | Flow Type | ACME | 113 |
| 92 | Flow Input Realm | ACME | 114 |
| 93 | Flow Input Src Addr | ACME | 115 |
| 94 | Flow Input Src Port | ACME | 116 |
| 95 | Flow Input Dest Address | ACME | 117 |
| 96 | Flow Input Dest Port | ACME | 118 |
| 97 | Flow Output Realm | ACME | 119 |
| 98 | Flow Output Src Address | ACME | 120 |
| 99 | Flow Output Src Port | ACME | 121 |
| 100 | Flow Output Dest Addr | ACME | 122 |
| 101 | Flow Output Dest Port | ACME | 123 |
| 102 | RTCP Called Packets Lost | ACME | 126 |
| 103 | RTCP Called Avg Jitter (msec) | ACME | 127 |
| 104 | RTCP Called Avg Latency (msec) | ACME | 128 |
| 105 | RTCP Called Max Jitter (msec) | ACME | 129 |
| 106 | RTCP Called Max Latency (msec) | ACME | 130 |
| 107 | RTP Called Packets Lost | ACME | 131 |
| 108 | RTP Called Avg Jitter (msec) | ACME | 132 |
| 109 | RTP Called Max Jitter (msec) | ACME | 133 |
| 110 | RTP Called Octets | ACME | 124 |
| 111 | RTP Called Packets | ACME | 125 |
| 112 | Charging Vector ICID | ACME | 54 |
| 113 | Charging Function Address | ACME | 55 |
| 114 | Firmware Version | ACME | 56 |
| 115 | Local timezone | ACME | 57 |
| 116 | Post Dial Delay (msec) | ACME | 58 |
| 117 | Primary routing Number | ACME | 64 |
| 118 | Originating Trunk Group | ACME | 65 |
| 119 | Terminating Trunk Group | ACME | 66 |
| 120 | Originating Trunk Context | ACME | 67 |
| 121 | Terminating Trunk Context | ACME | 68 |
| 122 | P Asserted ID | ACME | 69 |
| 123 | Ingress Local Address | ACME | 74 |
| 124 | Ingress Remote Address | ACME | 75 |
| 125 | Egress Local Address | ACME | 76 |
| 126 | Egress Remote Address | ACME | 77 |
| 127 | SIP DIVERSION | ACME | 70 |
| 128 | Intermediate Time | ACME | 63 |
| 129 | Accounting Session Time | none | 46 |
| 130 | Egress Routing Number | ACME | 134 |
| 131 | Ingress RPH | ACME | 135 |
| 132 | Egress RPH | ACME | 136 |

Table B-6 (Cont.) Local CDR Interim Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 133 | Diam-Session-Id | ACME | 172 |
| 134 | Sip-Method | ACME | 173 |
| 135 | Event-Time | ACME | 176 |
| 136 | Content-Disposition | ACME | 4119 |
| 137 | Originator | ACME | 4120 |
| 138 | Subscription-ID-Data | ACME | 4110 |
| 139 | Subscription-ID-Type | ACME | 4111 |
| 140 | Service-Context-ID | ACME | 4121 |
| 141 | Called-Asserted-Identity | ACME | 4117 |
| 142 | SDP-Media | ACME | 177 |
| 143 | SDP-Type | ACME | 4114 |
| 144 | Served-Party-IP | ACME | 4098 |
| 145 | User-Name | ACME | 4097 |
| 146 | Node-Func | ACME | 4100 |
| 147 | ApplicationId | ACME | 4101 |
| 148 | Role-Of-Node | ACME | 4102 |
| 149 | SIP-Request-Timestamp | ACME | 4124 |
| 150 | SIP-Request-Timestamp-Fraction | ACME | 4112 |
| 151 | SIP-Response-Timestamp | ACME | 4125 |
| 152 | SIP-Response-Timestamp-Fraction | ACME | 4113 |
| 153 | Access-Network-Information | ACME | 248 |
| 154 | Custom VSA 200 | ACME | 200 |
| 155 | Custom VSA 201 | ACME | 201 |
| 156 | Custom VSA 202 | ACME | 202 |
| 157 | Custom VSA 203 | ACME | 203 |
| 158 | Custom VSA 204 | ACME | 204 |
| 159 | Custom VSA 205 | ACME | 205 |
| 160 | Custom VSA 206 | ACME | 206 |
| 161 | Custom VSA 207 | ACME | 207 |
| 162 | Custom VSA 208 | ACME | 208 |
| 163 | Custom VSA 209 | ACME | 209 |
| 164 | Custom VSA 210 | ACME | 210 |
| 165 | Custom VSA 211 | ACME | 211 |
| 166 | Custom VSA 212 | ACME | 212 |
| 167 | Custom VSA 213 | ACME | 213 |
| 168 | Custom VSA 214 | ACME | 214 |
| 169 | Custom VSA 215 | ACME | 215 |
| 170 | Custom VSA 216 | ACME | 216 |
| 171 | Custom VSA 217 | ACME | 217 |
| 172 | Custom VSA 218 | ACME | 218 |
| 173 | Custom VSA 219 | ACME | 219 |
| 174 | Custom VSA 220 | ACME | 220 |
| 175 | Custom VSA 221 | ACME | 221 |
| 176 | Custom VSA 222 | ACME | 222 |

Table B-6 (Cont.) Local CDR Interim Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------------------|--------|------------|
| 177 | Custom VSA 223 | ACME | 223 |
| 178 | Custom VSA 224 | ACME | 224 |
| 179 | Custom VSA 225 | ACME | 225 |
| 180 | Custom VSA 226 | ACME | 226 |
| 181 | Custom VSA 227 | ACME | 227 |
| 182 | Custom VSA 228 | ACME | 228 |
| 183 | Custom VSA 229 | ACME | 229 |
| 184 | Custom VSA 230 | ACME | 230 |
| 185 | Calling-Media-Stop-Time | ACME | 231 |
| 186 | Called-Media-Stop-Time | ACME | 232 |
| 187 | Calling-Media-Stop-Time | ACME | 233 |
| 188 | Called-Media-Stop-Time | ACME | 234 |
| 189 | Flow Media Type | ACME | 142 |
| 190 | Flow Media Type | ACME | 143 |
| 191 | Flow Media Type | ACME | 144 |
| 192 | Flow Media Type | ACME | 145 |
| 193 | PGW-IP Address | ACME | 249 |
| 194 | SGW-IP Address | ACME | 249 |
| 195 | ORIG-IOI | ACME | 249 |
| 196 | TERM-IOI | ACME | 249 |
| 197 | IMEI | ACME | 249 |
| 198 | Node Functionality | ACME | 249 |
| 199 | History Info | ACME | 250 |
| 200 | P-Visited Network ID | ACME | 251 |
| 201 | IMSI | ACME | 252 |
| 202 | Reason-Header | none | 3401 |
| 203 | CDR Sequence Number | ACME | 59 |
| 204 | Origin-Realm | ACME | 4103 |
| 205 | Origin-Host | ACME | 4104 |
| 206 | Destination-Realm | ACME | 4105 |
| 207 | Destination-Host | ACME | 4106 |
| 208 | Stir-Signed-Request | ACME | 249 |
| 209 | Stir-Signed-Request-Exception-Id | ACME | 249 |
| 210 | Stir-Verified-Request | ACME | 249 |
| 211 | Stir-Verified-Request-Exception-Id | ACME | 249 |
| 212 | Stir-VS-Verstat | ACME | 249 |
| 213 | Stir-VS-Reason | ACME | 249 |
| 214 | Stir-TN-Used-For-AS-VS-Request | ACME | 249 |
| 215 | Stir-Div-Signed-Request | ACME | 249 |
| 216 | Stir-Div-Verified-Request | ACME | 249 |
| 217 | History Info2 | ACME | 253 |
| 218 | Stir-VS-Invite-State | ACME | 249 |
| 219 | Acme-SIP-HDR-TO | ACME | 249 |
| 220 | Stir-VS-attest | ACME | 178 |

Table B-6 (Cont.) Local CDR Interim Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|--|--------|------------|
| 221 | Stir-Div-VS-Reason | ACME | 179 |
| 222 | Stir-Div-VS-Verstat | ACME | 189 |
| 223 | Stir-AS-Bypass-Token | ACME | 181 |
| 224 | Number-Portability-Routing-Information | ACME | 2024 |
| 232 | VBD Tone Detected | ACME | 182 |

Local CDR Stop Record (Diameter)

The following table contains the local CDR layout, inclusive of all VSA data for a start record, when Diameter is selected as the account protocol.

Table B-7 Local CDR Stop Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|------------------------------|--------|------------|
| 1 | Accounting Status | none | 40 |
| 2 | NAS IP Address | none | 4 |
| 3 | NAS Port | none | 5 |
| 4 | Accounting Session ID | none | 44 |
| 5 | Ingress Session ID | ACME | 3 |
| 6 | Egress Session ID | ACME | 4 |
| 7 | Session Protocol Type | ACME | 43 |
| 8 | Session-Forked-Call-Id | ACME | 171 |
| 9 | Generic ID | ACME | 40 |
| 10 | Calling Station ID | none | 31 |
| 11 | Called Station ID | none | 30 |
| 12 | Accounting Termination Cause | none | 49 |
| 13 | Accounting Session Time | none | 46 |
| 14 | Cisco Setup Time | CISCO | 25 |
| 15 | Cisco Connect Time | CISCO | 28 |
| 16 | Cisco Disconnect Time | CISCO | 29 |
| 17 | Cisco Disconnect Cause | CISCO | 30 |
| 18 | Egress Network Interface ID | ACME | 139 |
| 19 | Egress Vlan Tag Value | ACME | 140 |
| 20 | Ingress Network Interface ID | ACME | 137 |
| 21 | Ingress Vlan Tag Value | ACME | 138 |
| 22 | Egress Realm | ACME | 42 |
| 23 | Ingress Realm | ACME | 41 |
| 24 | Flow Identifier | ACME | 1 |
| 25 | Flow Type | ACME | 2 |
| 26 | Flow Input Realm | ACME | 10 |
| 27 | Flow Input Src Addr | ACME | 11 |
| 28 | Flow Input Src Port | ACME | 12 |
| 29 | Flow Input Dest Address | ACME | 13 |

Table B-7 (Cont.) Local CDR Stop Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 30 | Flow Input Dest Port | ACME | 14 |
| 31 | Flow Output Realm | ACME | 20 |
| 32 | Flow Output Src Address | ACME | 21 |
| 33 | Flow Output Src Port | ACME | 22 |
| 34 | Flow Output Dest Addr | ACME | 23 |
| 35 | Flow Output Dest Port | ACME | 24 |
| 36 | RTCP Calling Packets Lost | ACME | 32 |
| 37 | RTCP Calling Avg Jitter (msec) | ACME | 33 |
| 38 | RTCP Calling Avg Latency (msec) | ACME | 34 |
| 39 | RTCP Calling Max Jitter (msec) | ACME | 35 |
| 40 | RTCP Calling Max Latency (msec) | ACME | 36 |
| 41 | RTP Calling Packets Lost | ACME | 37 |
| 42 | RTP Calling Avg Jitter (msec) | ACME | 38 |
| 43 | RTP Calling Max Jitter (msec) | ACME | 39 |
| 44 | RTP Calling Octets | ACME | 28 |
| 45 | RTP Calling Packets | ACME | 29 |
| 46 | Calling R-Factor | ACME | 151 |
| 47 | Calling MOS | ACME | 152 |
| 48 | Flow Identifier | ACME | 78 |
| 49 | Flow Type | ACME | 79 |
| 50 | Flow Input Realm | ACME | 80 |
| 51 | Flow Input Src Addr | ACME | 81 |
| 52 | Flow Input Src Port | ACME | 82 |
| 53 | Flow Input Dest Address | ACME | 83 |
| 54 | Flow Input Dest Port | ACME | 84 |
| 55 | Flow Output Realm | ACME | 85 |
| 56 | Flow Output Src Address | ACME | 86 |
| 57 | Flow Output Src Port | ACME | 87 |
| 58 | Flow Output Dest Addr | ACME | 88 |
| 59 | Flow Output Dest Port | ACME | 89 |
| 60 | RTCP Called Packets Lost | ACME | 46 |
| 61 | RTCP Called Avg Jitter (msec) | ACME | 47 |
| 62 | RTCP Called Avg Latency (msec) | ACME | 48 |
| 63 | RTCP Called Max Jitter (msec) | ACME | 49 |
| 64 | RTCP Called Max Latency (msec) | ACME | 50 |
| 65 | RTP Called Packets Lost | ACME | 51 |
| 66 | RTP Called Avg Jitter (msec) | ACME | 52 |
| 67 | RTP Called Max Jitter (msec) | ACME | 53 |
| 68 | RTP Called Octets | ACME | 44 |
| 69 | RTP Called Packets | ACME | 45 |
| 70 | Called R-Factor | ACME | 153 |
| 71 | Called MOS | ACME | 154 |
| 72 | Flow Identifier | ACME | 90 |
| 73 | Flow Type | ACME | 91 |

Table B-7 (Cont.) Local CDR Stop Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 74 | Flow Input Realm | ACME | 92 |
| 75 | Flow Input Src Addr | ACME | 93 |
| 76 | Flow Input Src Port | ACME | 94 |
| 77 | Flow Input Dest Address | ACME | 95 |
| 78 | Flow Input Dest Port | ACME | 96 |
| 79 | Flow Output Realm | ACME | 97 |
| 80 | Flow Output Src Address | ACME | 98 |
| 81 | Flow Output Src Port | ACME | 99 |
| 82 | Flow Output Dest Addr | ACME | 100 |
| 83 | Flow Output Dest Port | ACME | 101 |
| 84 | RTCP Calling Packets Lost | ACME | 104 |
| 85 | RTCP Calling Avg Jitter (msec) | ACME | 105 |
| 86 | RTCP Calling Avg Latency (msec) | ACME | 106 |
| 87 | RTCP Calling Max Jitter (msec) | ACME | 107 |
| 88 | RTCP Calling Max Latency (msec) | ACME | 108 |
| 89 | RTP Calling Packets Lost | ACME | 109 |
| 90 | RTP Calling Avg Jitter (msec) | ACME | 110 |
| 91 | RTP Calling Max Jitter (msec) | ACME | 111 |
| 92 | RTP Calling Octets | ACME | 102 |
| 93 | RTP Calling Packets | ACME | 103 |
| 94 | Flow Identifier | ACME | 112 |
| 95 | Flow Type | ACME | 113 |
| 96 | Flow Input Realm | ACME | 114 |
| 97 | Flow Input Src Addr | ACME | 115 |
| 98 | Flow Input Src Port | ACME | 116 |
| 99 | Flow Input Dest Address | ACME | 117 |
| 100 | Flow Input Dest Port | ACME | 118 |
| 101 | Flow Output Realm | ACME | 119 |
| 102 | Flow Output Src Address | ACME | 120 |
| 103 | Flow Output Src Port | ACME | 121 |
| 104 | Flow Output Dest Addr | ACME | 122 |
| 105 | Flow Output Dest Port | ACME | 123 |
| 106 | RTCP Called Packets Lost | ACME | 126 |
| 107 | RTCP Called Avg Jitter (msec) | ACME | 127 |
| 108 | RTCP Called Avg Latency (msec) | ACME | 128 |
| 109 | RTCP Called Max Jitter (msec) | ACME | 129 |
| 110 | RTCP Called Max Latency (msec) | ACME | 130 |
| 111 | RTP Called Packets Lost | ACME | 131 |
| 112 | RTP Called Avg Jitter (msec) | ACME | 132 |
| 113 | RTP Called Max Jitter (msec) | ACME | 133 |
| 114 | RTP Called Octets | ACME | 124 |
| 115 | RTP Called Packets | ACME | 125 |
| 116 | Charging Vector ICID | ACME | 54 |
| 117 | Charging Function Address | ACME | 55 |

Table B-7 (Cont.) Local CDR Stop Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 118 | Firmware Version | ACME | 56 |
| 119 | Local timezone | ACME | 57 |
| 120 | Post Dial Delay (msec) | ACME | 58 |
| 121 | Primary routing Number | ACME | 64 |
| 122 | Originating Trunk Group | ACME | 65 |
| 123 | Terminating Trunk Group | ACME | 66 |
| 124 | Originating Trunk Context | ACME | 67 |
| 125 | Terminating Trunk Context | ACME | 68 |
| 126 | P Asserted ID | ACME | 69 |
| 127 | Ingress Local Address | ACME | 74 |
| 128 | Ingress Remote Address | ACME | 75 |
| 129 | Egress Local Address | ACME | 76 |
| 130 | Egress Remote Address | ACME | 77 |
| 131 | SIP DIVERSION | ACME | 70 |
| 132 | Session Disposition | ACME | 60 |
| 133 | Disconnect Initiator | ACME | 61 |
| 134 | Disconnect Cause | ACME | 62 |
| 135 | Sip Status Code | ACME | 71 |
| 136 | Egress Routing Number | ACME | 134 |
| 137 | Ingress RPH | ACME | 135 |
| 138 | Egress RPH | ACME | 136 |
| 139 | refer call transfer ID | ACME | 141 |
| 140 | Diam-Session-Id | ACME | 172 |
| 141 | Sip-Method | ACME | 173 |
| 142 | Event-Time | ACME | 176 |
| 143 | Content-Disposition | ACME | 4119 |
| 144 | Originator | ACME | 4120 |
| 145 | Subscription-ID-Data | ACME | 4110 |
| 146 | Subscription-ID-Type | ACME | 4111 |
| 147 | Service-Context-ID | ACME | 4121 |
| 148 | Called-Asserted-Identity | ACME | 4117 |
| 149 | SDP-Media | ACME | 177 |
| 150 | SDP-Type | ACME | 4114 |
| 151 | Served-Party-IP | ACME | 4098 |
| 152 | User-Name | ACME | 4097 |
| 153 | Node-Func | ACME | 4100 |
| 154 | ApplicationId | ACME | 4101 |
| 155 | Role-Of-Node | ACME | 4102 |
| 156 | SIP-Request-Timestamp | ACME | 4124 |
| 157 | SIP-Request-Timestamp-Fraction | ACME | 4112 |
| 158 | SIP-Response-Timestamp | ACME | 4125 |
| 159 | SIP-Response-Timestamp-Fraction | ACME | 4113 |
| 160 | Access-Network-Information | ACME | 248 |
| 161 | Cause-Code | ACME | 4099 |

Table B-7 (Cont.) Local CDR Stop Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|---------------------------------|--------|------------|
| 162 | Custom VSA 200 | ACME | 200 |
| 163 | Custom VSA 201 | ACME | 201 |
| 164 | Custom VSA 202 | ACME | 202 |
| 165 | Custom VSA 203 | ACME | 203 |
| 166 | Custom VSA 204 | ACME | 204 |
| 167 | Custom VSA 205 | ACME | 205 |
| 168 | Custom VSA 206 | ACME | 206 |
| 169 | Custom VSA 207 | ACME | 207 |
| 170 | Custom VSA 208 | ACME | 208 |
| 171 | Custom VSA 209 | ACME | 209 |
| 172 | Custom VSA 210 | ACME | 210 |
| 173 | Custom VSA 211 | ACME | 211 |
| 174 | Custom VSA 212 | ACME | 212 |
| 175 | Custom VSA 213 | ACME | 213 |
| 176 | Custom VSA 214 | ACME | 214 |
| 177 | Custom VSA 215 | ACME | 215 |
| 178 | Custom VSA 216 | ACME | 216 |
| 179 | Custom VSA 217 | ACME | 217 |
| 180 | Custom VSA 218 | ACME | 218 |
| 181 | Custom VSA 219 | ACME | 219 |
| 182 | Custom VSA 220 | ACME | 220 |
| 183 | Custom VSA 221 | ACME | 221 |
| 184 | Custom VSA 222 | ACME | 222 |
| 185 | Custom VSA 223 | ACME | 223 |
| 186 | Custom VSA 224 | ACME | 224 |
| 187 | Custom VSA 225 | ACME | 225 |
| 188 | Custom VSA 226 | ACME | 226 |
| 189 | Custom VSA 227 | ACME | 227 |
| 190 | Custom VSA 228 | ACME | 228 |
| 191 | Custom VSA 229 | ACME | 229 |
| 192 | Custom VSA 230 | ACME | 230 |
| 193 | Calling-Media-Stop-Time | ACME | 231 |
| 194 | Called-Media-Stop-Time | ACME | 232 |
| 195 | Calling-Media-Stop-Time | ACME | 233 |
| 196 | Called-Media-Stop-Time | ACME | 234 |
| 197 | Flow Media Type | ACME | 142 |
| 198 | Flow Media Type | ACME | 143 |
| 199 | Flow Media Type | ACME | 144 |
| 200 | Flow Media Type | ACME | 145 |
| 201 | RTP Calling Octets Transmitted | ACME | 240 |
| 202 | RTP Calling Packets Transmitted | ACME | 241 |
| 203 | RTP Called Octets Transmitted | ACME | 242 |
| 204 | RTP Called Packets Transmitted | ACME | 243 |
| 205 | RTP Calling Octets Transmitted | ACME | 244 |

Table B-7 (Cont.) Local CDR Stop Record (Diameter)

| CSV Position | Attribute Name | Vendor | CDR Number |
|--------------|--|--------|------------|
| 206 | RTP Calling Packets Transmitted | ACME | 245 |
| 207 | RTP Calling Octets Transmitted | ACME | 244 |
| 208 | RTP Calling Packets Transmitted | ACME | 245 |
| 209 | MSRP Called Octets | ACME | 249 |
| 210 | MSRP Called Packets | ACME | 249 |
| 211 | MSRP Called Octets Transmitted | ACME | 249 |
| 212 | MSRP Called Packets Transmitted | ACME | 249 |
| 213 | MSRP Calling Octets | ACME | 249 |
| 214 | MSRP Calling Packets | ACME | 249 |
| 215 | MSRP Calling Octets Transmitted | ACME | 249 |
| 216 | MSRP Calling Packets Transmitted | ACME | 249 |
| 217 | PGW-IP Address | ACME | 249 |
| 218 | SGW-IP Address | ACME | 249 |
| 219 | ORIG-IOI | ACME | 249 |
| 220 | TERM-IOI | ACME | 249 |
| 221 | IMEI | ACME | 249 |
| 222 | Node Functionality | ACME | 249 |
| 223 | History Info | ACME | 250 |
| 224 | P-Visited Network ID | ACME | 251 |
| 225 | IMSI | ACME | 252 |
| 226 | Reason-Header | none | 3401 |
| 227 | CDR Sequence Number | ACME | 59 |
| 228 | Origin-Realm | ACME | 4103 |
| 229 | Origin-Host | ACME | 4104 |
| 230 | Destination-Realm | ACME | 4105 |
| 231 | Destination-Host | ACME | 4106 |
| 232 | Stir-Signed-Request | ACME | 249 |
| 233 | Stir-Signed-Request-Exception-Id | ACME | 249 |
| 234 | Stir-Verified-Request | ACME | 249 |
| 235 | Stir-Verified-Request-Exception-Id | ACME | 249 |
| 236 | Stir-VS-Verstat | ACME | 249 |
| 237 | Stir-VS-Reason | ACME | 249 |
| 238 | Stir-TN-Used-For-AS-VS-Request | ACME | 249 |
| 239 | Stir-Div-Signed-Request | ACME | 249 |
| 240 | Stir-Div-Verified-Request | ACME | 249 |
| 241 | History Info2 | ACME | 253 |
| 242 | Stir-VS-Invite-State | ACME | 249 |
| 243 | Acme-SIP-HDR-TO | ACME | 249 |
| 244 | Stir-VS-attest | ACME | 178 |
| 245 | Stir-Div-VS-Reason | ACME | 179 |
| 246 | Stir-Div-VS-Verstat | ACME | 180 |
| 247 | Stir-AS-Bypass-Token | ACME | 181 |
| 248 | Number-Portability-Routing-Information | ACME | 2024 |
| 256 | VBD Tone Detected | ACME | 182 |

Local CDR Message Record for SMS (Diameter)

The SBC creates an Event CSV file for SMS messages when the generate-event = message parameter is enabled. The following table shows the inclusive CSV element order, when protocol = diameter.

Table B-8 Local CDR Message Record (Diameter)

| CSV Placement | Attribute Name | Vendor | CDR Number |
|---------------|---------------------------------|--------|------------|
| 1 | Accounting Status | none | 40 |
| 2 | NAS IP Address | none | 4 |
| 3 | NAS Port | none | 5 |
| 4 | Accounting Session ID | none | 44 |
| 5 | Ingress Session ID | ACME | 3 |
| 6 | Egress Session ID | ACME | 4 |
| 7 | Session Protocol Type | ACME | 43 |
| 8 | Session-Forked-Call-Id | ACME | 171 |
| 9 | Generic ID | ACME | 40 |
| 10 | Calling Station ID | none | 31 |
| 11 | Called Station ID | none | 30 |
| 12 | Egress Network Interface ID | ACME | 137 |
| 13 | Egress Vlan Tag Value | ACME | 140 |
| 14 | Ingress Network Interface ID | ACME | 137 |
| 15 | Ingress Vlan Tag Value | ACME | 138 |
| 16 | Cisco Setup Time | CISCO | 25 |
| 17 | Cisco Connect Time | CISCO | 28 |
| 18 | Diam-Session-Id | ACME | 172 |
| 19 | Sip-Method | ACME | 173 |
| 20 | Event-Time | ACME | 176 |
| 21 | Content-Disposition | ACME | 4119 |
| 22 | Originator | ACME | 4120 |
| 23 | Subscription-ID-Data | ACME | 4110 |
| 24 | Subscription-ID-Type | ACME | 4111 |
| 25 | Service-Context-ID | ACME | 4121 |
| 26 | Called-Asserted-Identity | ACME | 4117 |
| 27 | User-Name | ACME | 4097 |
| 28 | Node-Func | ACME | 4100 |
| 29 | ApplicationId | ACME | 4101 |
| 30 | Role-Of-Node | ACME | 4102 |
| 31 | SIP-Request-Timestamp | ACME | 4124 |
| 32 | SIP-Request-Timestamp-Fraction | ACME | 4112 |
| 33 | SIP-Response-Timestamp | ACME | 4125 |
| 34 | SIP-Response-Timestamp-Fraction | ACME | 4112 |
| 35 | Access-Network-Information | ACME | 248 |
| 36 | Event | ACME | |
| 37 | Expires | ACME | |
| 38 | Associated-URI | ACME | |

Table B-8 (Cont.) Local CDR Message Record (Diameter)

| CSV Placement | Attribute Name | Vendor | CDR Number |
|---------------|--------------------------|--------|------------|
| 39 | Cause-Code | ACME | |
| 40 | Custom VSA 200 | ACME | 200 |
| 41 | Custom VSA 201 | ACME | 201 |
| 42 | Custom VSA 202 | ACME | 202 |
| 43 | Custom VSA 203 | ACME | 203 |
| 44 | Custom VSA 204 | ACME | 204 |
| 45 | Custom VSA 205 | ACME | 205 |
| 46 | Custom VSA 206 | ACME | 206 |
| 47 | Custom VSA 207 | ACME | 207 |
| 48 | Custom VSA 208 | ACME | 208 |
| 49 | Custom VSA 209 | ACME | 209 |
| 50 | Custom VSA 210 | ACME | 210 |
| 51 | Custom VSA 211 | ACME | 211 |
| 52 | Custom VSA 212 | ACME | 212 |
| 53 | Custom VSA 213 | ACME | 213 |
| 54 | Custom VSA 214 | ACME | 214 |
| 55 | Custom VSA 215 | ACME | 215 |
| 56 | Custom VSA 216 | ACME | 216 |
| 57 | Custom VSA 217 | ACME | 217 |
| 58 | Custom VSA 218 | ACME | 218 |
| 59 | Custom VSA 219 | ACME | 219 |
| 60 | Custom VSA 220 | ACME | 220 |
| 61 | Custom VSA 221 | ACME | 221 |
| 62 | Custom VSA 222 | ACME | 222 |
| 63 | Custom VSA 223 | ACME | 223 |
| 64 | Custom VSA 224 | ACME | 224 |
| 65 | Custom VSA 225 | ACME | 225 |
| 66 | Custom VSA 226 | ACME | 226 |
| 67 | Custom VSA 227 | ACME | 227 |
| 68 | Custom VSA 228 | ACME | 228 |
| 69 | Custom VSA 229 | ACME | 229 |
| 70 | Custom VSA 230 | ACME | 230 |
| 71 | PGW-IP Address | ACME | 249 |
| 72 | SGW-IP Address | ACME | 249 |
| 73 | ORIG-IOI | ACME | 249 |
| 74 | TERM-IOI | ACME | 249 |
| 75 | IMEI | ACME | 249 |
| 76 | Node Functionality | ACME | 249 |
| 77 | Sms-Msg-Type | ACME | |
| 78 | Sms-Calling-Party-Number | ACME | |
| 79 | Sms-Called-Party-Number | ACME | |
| 80 | Sms-Msg-length | ACME | |
| 81 | P-Visited Network ID | ACME | 251 |
| 82 | IMSI | ACME | 252 |

Table B-8 (Cont.) Local CDR Message Record (Diameter)

| CSV Placement | Attribute Name | Vendor | CDR Number |
|---------------|---------------------|--------|------------|
| 83 | CDR Sequence Number | ACME | 59 |
| 84 | Origin-Realm | ACME | 4103 |
| 85 | Origin-Host | ACME | 4104 |
| 86 | Destination-Realm | ACME | 4105 |
| 87 | Destination-Host | ACME | 4106 |

C

Oracle Rf Interface Support

The SBC supports numerous AVPs in its Diameter-based Rf implementation. Currently AVPs belong to:

- The Diameter base AVPs found in RFC3588 and RFC4006.
- For 3GPP AVPs, if not specified by this document, their definition follows corresponding 3GPP specifications.
- Oracle proprietary Rf AVPs. Please see Acme-Packet-Specific-Extension-Rf AVP.

Diameter AVP Notation

3GPP 32.299 states the following symbols are used in the message format definitions:

<AVP> indicates a mandatory AVP with a fixed position in the message.

{AVP} indicates a mandatory AVP in the message.

[AVP] indicates an optional AVP in the message.

*AVP indicates that multiple occurrences of an AVP is possible.

Table Explanation

Each row in the following AVP tables contain:

- AVP Name
- AVP Number
- Reference where the AVP was defined
- Valid appearance in start, interim, stop, or event records
- For grouped AVPs, link to the group's respective section.

Root ACR Message Format

The following table contains the top level AVPs that may be present in an SBC-generated message.

| AVP | Number | Reference | Start | Interim | Stop | Event | Grouped |
|------------------------------|--------|-----------|-------|---------|------|-------|---------|
| { Session-Id } | 263 | Base | Y | Y | Y | Y | N/A |
| { Origin-Host } | 264 | Base | Y | Y | Y | Y | N/A |
| { Origin-Realm } | 296 | Base | Y | Y | Y | Y | N/A |
| { Destination-Realm } | 283 | Base | Y | Y | Y | Y | N/A |
| { Accounting-Record-Type } | 480 | Base | Y | Y | Y | Y | N/A |
| { Accounting-Record-Number } | 485 | Base | Y | Y | Y | Y | N/A |

| AVP | Number | Reference | Start | Interim | Stop | Event | Grouped |
|---------------------------------------|--------|-----------|-------|---------|------|-------|---------------------------------------|
| [Acct-Application-Id] | 259 | Base | Y | Y | Y | Y | N/A |
| [User-Name] | 1 | Base | Y | Y | Y | Y | N/A |
| [Origin-state-ID] | 278 | Base | Y | Y | Y | Y | N/A |
| [Event-Timestamp] | 55 | Base | Y | Y | Y | Y | N/A |
| [Service-Context-ID] | 461 | 3GPP | Y | Y | Y | Y | N/A |
| [Service-Information] | 873 | 3GPP | Y | Y | Y | Y | Service-Information AVP |
| [Acme-Packet-Specific-Extension-Rf] | 1 | ACME | Y | Y | Y | | Acme-Packet-Specific-Extension-Rf AVP |

Service Information AVP

The Service-Information AVP (AVP code 873) is of type Grouped.

| AVP | Number | Reference | Start | Interim | Stop | Event | Grouped |
|---------------------|--------|-----------|-------|---------|------|-------|---------------------|
| [Subscription-ID] | 443 | 3GPP | Y | Y | Y | | Subscription ID AVP |
| [IMS Information] | 876 | 3GPP | Y | Y | Y | Y | IMS Information AVP |

Subscription ID AVP

The Subscription ID AVP (AVP code 108) contains the identification of the user that is going to access the service in order to be identified by the OCS.

| AVP | Number | Reference | Start | Interim | Stop | Event | Event |
|--------------------------|--------|-----------|-------|---------|------|-------|-------|
| [Subscription-ID-Data] | 444 | 3GPP | Y | Y | Y | Y | N/A |
| [Subscription-ID-Type] | 450 | 3GPP | Y | Y | Y | Y | N/A |

IMS Information AVP

The IMS-Information AVP (AVP code 876) is of type Grouped. Its purpose is to allow the transmission of additional IMS service specific information elements.

| AVP | Number | Reference | Start | Interim | Stop | Event | Group |
|----------------------|--------|-----------|-------|---------|------|-------|----------------|
| [Event-Type] | 823 | 3GPP | Y | Y | Y | Y | Event-Type AVP |
| [Role-of-Node] | 829 | 3GPP | Y | Y | Y | Y | N/A |
| {Node-Functionality} | 862 | 3GPP | Y | Y | Y | Y | N/A |
| [User-Session-Id] | 830 | 3GPP | Y | Y | Y | N/A | N/A |

| AVP | Number | Reference | Start | Interim | Stop | Event | Group |
|--|--------|-----------|-------|---------|------|-------|-------------------------------|
| * [Calling-Party-Address] | 831 | 3GPP | Y | Y | Y | Y | N/A |
| [Called-Party-Address] | 832 | 3GPP | Y | Y | Y | Y | N/A |
| * [Called-Asserted-Identity] | 1250 | 3GPP | Y | N/A | N/A | N/A | N/A |
| * [Associated-URI] | 856 | 3GPP | N/A | N/A | N/A | Y | N/A |
| [Time-Stamps] | 833 | 3GPP | Y | Y | Y | Y | Time Stamps AVP |
| [Inter-Operator-Identifier] | 838 | 3GPP | Y | Y | Y | N/A | Inter-Operator-Identifier AVP |
| *[SDP-Session-Description] | 842 | 3GPP | Y | Y | N/A | N/A | N/A |
| *[SDP-Media-Component] | 843 | 3GPP | Y | Y | N/A | N/A | SDP-Media-Component AVP |
| [IMS-Charging-Identifier] | 841 | 3GPP | Y | Y | Y | N/A | N/A |
| *[Early-Media-Description] | 1272 | 3GPP | Y | N/A | N/A | Y | Early-Media-Description AVP |
| *[Message-Body] | 889 | 3GPP | Y | Y | Y | Y | Message-Body AVP |
| [Served-Party-IP-Address] | 848 | 3GPP | Y* | Y* | Y* | N/A | N/A |
| [Access-Network-Information] | 1263 | N/A | Y | Y | Y | Y | N/A |
| [Cause-Code] | 861 | N/A | N/A | N/A | Y | Y | N/A |
| [Reason-Header] | 3401 | Base | N | N | Y | Y | N/A |
| [Number-Portability-Routing-Information] | 2024 | 3GPP | Y** | Y** | Y** | N | N/A |

Y*—Served-Party-IP-Address appears if **sip-interface**, **sip-ims-feature** is set to **enabled**.

Y**—Number-Portability-Routing-Information appears if **number-port-routing-infn-avp** is configured.

Reason-Header AVP (3401)

The Reason-Header AVP (3401), is a UTF8 string that contains the content of the Reason-header detected by the SBC in SIP BYE, CANCEL, and SIP error responses. It may contain multiple entries if the system detects multiple reason headers. The system includes this AVP in the ACR for Accounting-Record-Type [STOP/EVENT] when there is an active **account-config** running diameter, and you have enabled the **add-reason-header** parameter in the **sip-config**.

The SBC expects this AVP in an ACR message, as follows follow.

| AVP | Number | AVP Type | Reference | Start | Interim | Stop | Event |
|-----------------------|--------|-------------|-----------|-------|---------|------|-------|
| [Reason-Header AVP] | 3401 | UTF8 String | Base | N | N | Y | Y |

The syntax of the Reason-Header AVP is:

```
AVP: Reason-Header(3401) l=49 f=VM- vnd=TGPP val=Q.850;cause=16;text="Call
Terminated"
AVP Code: 3401 Reason-Header
AVP Flags: 0xc0, Vendor-Specific: Set, Mandatory: Set
AVP Length: 49
AVP Vendor Id: 3GPP (10415)
Reason-Header: Q.850;cause=16;text="Call Terminated"
Padding: 000000
```

- AVP: Reason-Header(3401) l=49 f=VM- vnd=TGPP val=Q.850;cause=16;text="Call Terminated"
- AVP Code: 3401 Reason-Header
- AVP Flags: 0xc0,
- Vendor-Specific: Set, Mandatory: Set
- AVP Length: 49
- AVP Vendor Id: 3GPP (10415)
- Reason-Header: Q.850;cause=16;text="Call Terminated"
- Padding: 000000

Number-Portability-Routing-Information (2024)

This AVPs is a vendor-specific UTF8 string (D_AVP_NUMBER_PORT_ROUTING_INF), which is populated with the value of **rn** from the Request URI header received in an initial SIP INVITE or a mobile number portability (MNP) response from an ENUM server. It is included in the IMS-Information AVP (AVP code 876) grouped AVP.

This AVP uses the following ABNF grammar.

```
< Acme-Number-Portability-Routing-Information>:: = < AVP Header: 2024 >
[AVP_ACME_NUMBER_PORTABILITY_ROUTING_INFORMATION]
```

By default, the SBC does not populate this AVP. You enable it by configuring the **number-port-routing-infn-avp** option in the **account-config**:

```
ORACLE(account-config)#options +number-port-routing-infn-avp
```

Event-Type AVP

The Event-Type AVP (AVP code 823) is of type Grouped and contains information about the type of chargeable telecommunication service/event for which the accounting-request and/or credit control request message(s) is generated.

| AVP | Number | Acme # | Reference | Start | Interim | Stop | Event |
|----------------|--------|--------|-----------|-------|---------|------|-------|
| [SIP-Method] | 824 | 173 | 3GPP | Y | Y | Y | Y |
| [Event] | 825 | 245 | 3GPP | N/A | N/A | N/A | Y |
| [Expires] | 888 | 246 | 3GPP | N/A | N/A | N/A | Y |

Time Stamps AVP

The Time-Stamps AVP (AVP code 833) is of type Grouped and holds the time of the initial SIP request and the time of the response to the initial SIP Request.

| AVP | Number | Reference | Start | Interim | Stop | Event |
|-------------------------------------|--------|-----------|-------|---------|------|-------|
| [SIP-Request-Timestamp] | 834 | 3GPP | Y | Y | Y | Y |
| [SIP-Response-Timestamp] | 835 | 3GPP | Y* | Y | Y | Y |
| [SIP-Request-Timestamp-Fraction] | 2301 | 3GPP | Y | Y | Y | Y |
| [SIP-Response-Timestamp-Fraction] | 2302 | 3GPP | Y* | Y | Y | Y |

Y*—These AVPs appear in start records if **account-config, generate-start** is set to **OK**. If **generate-start=invite**, then they are not generated in the Start record.

Inter-Operator-Identifier AVP

The Inter-Operator-Identifier AVP (AVP code 838) is of type Grouped and holds the identification of the network neighbors (originating and terminating) as exchanged via SIP signalling.

| AVP | Number | Reference | Start | Interim | Stop | Event |
|---------------------|--------|-----------|-------|---------|------|-------|
| [Originating-IOI] | 839 | 3GPP | Y | Y | Y | N/A |
| [Termination-IOI] | 840 | 3GPP | Y* | Y | Y | N/A |

Y*—These AVPs appear in start records if **account-config, generate-start** is set to **OK**.

SDP-Media-Component AVP

The SDP- Media-Component AVP (AVP code 843) is of type Grouped and contains information about media used for a IMS session.

| AVP | Number | Reference | Start | Interim | Stop | Event |
|-----------------------------|--------|-----------|-------|---------|------|-------|
| [SDP-Media-Name] | 844 | 3GPP | Y | Y | N/A | N/A |
| * [SDP-Media-Description] | 845 | 3GPP | Y | Y | N/A | N/A |
| [SDP-Type] | 2036 | 3GPP | Y | Y | N/A | N/A |

Early-Media-Description AVP

The Early-Media-Description AVP (AVP code 1272) is of type grouped and describes the SDP session, media parameters and timestamps related to media components set to active according to SDP signalling exchanged during a SIP session establishment before the final successful or unsuccessful SIP answer to the initial SIP INVITE message is received. Once a media component has been set to active, subsequent status changes shall also be registered.

| AVP | Number | Reference | Start | Interim | Stop | Event | Group |
|--------------------|--------|-----------|-------|---------|------|-------|--------------------|
| [SDP-TimeStamps] | 1273 | 3GPP | Y | N/A | N/A | Y | SDP-Timestamps AVP |

| AVP | Number | Reference | Start | Interim | Stop | Event | Group |
|------------------------------|--------|-----------|-------|---------|------|-------|-------------------------|
| *[SDP-Session-Description] | 842 | 3GPP | Y | Y | N/A | N/A | N/A |
| *[SDP-Media-Component] | 843 | 3GPP | Y | Y | N/A | N/A | SDP-Media-Component AVP |

SDP-Timestamps AVP

The SDP-TimeStamps AVP (AVP code 1273) is of type Grouped and holds the time of the SDP offer and the SDP answer.

| AVP | Number | Reference | Start | Interim | Stop | Event |
|--------------------------|--------|-----------|-------|---------|------|-------|
| [SDP-Offer-Timestamp] | 1274 | N/A | Y | N/A | N/A | Y |
| [SDP-Answer-Timestamp] | 1275 | N/A | Y | N/A | N/A | Y |

Message-Body AVP

The Message-Body AVP (AVP Code 889) is of type Grouped AVP and holds information about the message bodies including user-to-user data.

| AVP | Number | Reference | Start | Interim | Stop | Event |
|-------------------------|--------|-----------|-------|---------|------|-------|
| { Content-Type } | 826 | 3GPP | Y | Y | Y | Y |
| { Content-Length } | 827 | 3GPP | Y | Y | Y | Y |
| [Content-Disposition] | 828 | 3GPP | Y | Y | Y | Y |
| [Originator] | 864 | 3GPP | Y | Y | Y | Y |

Acme-Packet-Specific-Extension-Rf AVP

The Oracle Acme-Packet-Specific-Extension-Rf AVP uses vendor ID 9148. The following section includes the ACME AVP descriptions.

| AVP | ACME Diameter Attribute | Start | Interim | Stop | Event | AVP Type |
|----------------------------------|-------------------------|-------|---------|------|-------|------------|
| Ingress-Realm | 2 | Y | Y | Y | N | UTF8String |
| Egress-Realm | 3 | Y | Y | Y | N | UTF8String |
| Ingress-TGRP | 4 | Y | Y | Y | N | UTF8String |
| Egress-TGRP | 5 | Y | Y | Y | N | UTF8String |
| Ingress-Trunk-Context | 6 | Y | Y | Y | N | UTF8String |
| Egress-Trunk-Context | 7 | N | Y | Y | N | UTF8String |
| Ingress-Local-Signaling-Address | 8 | Y | Y | Y | N | UTF8String |
| Ingress-Remote-Signaling-Address | 9 | Y | Y | Y | N | UTF8String |
| Egress-Local-Signaling-Address | 10 | Y | Y | Y | N | UTF8String |

| AVP | ACME Diameter Attribute | Start | Interim | Stop | Event | AVP Type |
|--|-------------------------|-------|---------|------|-------|------------|
| Egress-Remote-Signaling-Address | 11 | Y | Y | Y | N | UTF8String |
| Ingress-Local-Media-Address | 12 | Y | Y | Y | N | UTF8String |
| Ingress-Remote-Media-Address | 13 | Y | Y | Y | N | UTF8String |
| Egress-Local-Media-Address | 14 | Y | Y | Y | N | UTF8String |
| Egress-Remote-Media-Address | 15 | Y | Y | Y | N | UTF8String |
| NAS-Port | 16 | Y | Y | Y | N | Unsigned32 |
| Acme-Session-Ingress-CallID | 17 | Y | Y | Y | N | UTF8String |
| Acme-Session-Egress-CallID | 18 | Y | Y | Y | N | UTF8String |
| Acme-Session-Protocol-Type | 19 | Y | Y | Y | N | UTF8String |
| Acme-FlowID-FS1-F | 20 | Y | Y | Y | N | UTF8String |
| Acme-FlowType-FS1-F | 21 | Y | Y | Y | N | UTF8String |
| Ingress-Local-Media-Port | 22 | Y | Y | Y | N | Unsigned32 |
| Ingress-Remote-Media-Port | 23 | Y | Y | Y | N | Unsigned32 |
| Egress-Local-Media-Port | 24 | Y | Y | Y | N | Unsigned32 |
| Egress-Remote-Media-Port | 25 | Y | Y | Y | N | Unsigned32 |
| Acme-Session-Charging-Function-Address | 26 | Y | Y | Y | N | UTF8String |
| Acme-Local-Timezone | 27 | Y | Y | Y | N | UTF8String |
| Acme-Post-Dial-Delay | 28 | Y | Y | Y | N | Unsigned32 |
| Acme-SIP-Diversion | 29 | Y | Y | Y | N | UTF8String |
| Acme-Session-Disposition | 30 | N | N | Y | N | Unsigned32 |
| Disconnect-Initiator | 31 | N | N | Y | N | Unsigned32 |
| Terminate-Cause | 32 | N | N | Y | N | Unsigned32 |
| Acme-Session-Disconnect-Cause | 33 | N | N | Y | N | Unsigned32 |
| Acme-SIP-Status | 34 | N | N | Y | N | Integer32 |
| Acme-FlowType-FS1-R | 35 | N | Y | Y | N | UTF8String |
| Acme-Packet-Specific-Rf-QoS | 37 | N | N | Y | N | Grouped |

| AVP | ACME Diameter Attribute | Start | Interim | Stop | Event | AVP Type |
|-------------------------------------|-------------------------|-------|---------|------|-------|------------|
| RTP-Calling-Octets-FS1 | 38 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Octets-FS2 | 39 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Packets-FS1 | 40 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Packets-FS2 | 41 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Octets-Transmitted-FS1 | 42 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Octets-Transmitted-FS2 | 43 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Packets-Transmitted-FS1 | 44 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Packets-Transmitted-FS2 | 45 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Packets-Lost-FS1 | 46 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Packets-Lost-FS2 | 47 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Avg-Jitter-FS1 | 48 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Avg-Jitter-FS2 | 49 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Max-Jitter-FS1 | 50 | N | N | Y | N | Unsigned32 |
| RTP-Calling-Max-Jitter-FS2 | 51 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Packets-Lost-FS1 | 52 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Packets-Lost-FS2 | 53 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Avg-Jitter-FS1 | 54 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Avg-Jitter-FS2 | 55 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Avg-Latency-FS1 | 56 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Avg-Latency-FS2 | 57 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Max-Jitter-FS1 | 58 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Max-Jitter-FS2 | 59 | N | N | Y | N | Unsigned32 |
| RTCP-Calling-Max-Latency-FS1 | 60 | N | N | Y | N | Unsigned32 |

| AVP | ACME Diameter Attribute | Start | Interim | Stop | Event | AVP Type |
|------------------------------------|-------------------------|-------|---------|------|-------|------------|
| RTCP-Calling-Max-Latency-FS2 | 61 | N | N | Y | N | Unsigned32 |
| RTP-Called-Octets-FS1 | 62 | N | N | Y | N | Unsigned32 |
| RTP-Called-Octets-FS2 | 63 | N | N | Y | N | Unsigned32 |
| RTP-Called-Packets-FS1 | 64 | N | N | Y | N | Unsigned32 |
| RTP-Called-Packets-FS2 | 65 | N | N | Y | N | Unsigned32 |
| RTP-Called-Octets-Transmitted-FS1 | 66 | N | N | Y | N | Unsigned32 |
| RTP-Called-Octets-Transmitted-FS2 | 67 | N | N | Y | N | Unsigned32 |
| RTP-Called-Packets-Transmitted-FS1 | 68 | N | N | Y | N | Unsigned32 |
| RTP-Called-Packets-Transmitted-FS2 | 69 | N | N | Y | N | Unsigned32 |
| RTP-Called-Packets-Lost-FS1 | 70 | N | N | Y | N | Unsigned32 |
| RTP-Called-Packets-Lost-FS2 | 71 | N | N | Y | N | Unsigned32 |
| RTP-Called-Avg-Jitter-FS1 | 72 | N | N | Y | N | Unsigned32 |
| RTP-Called-Avg-Jitter-FS2 | 73 | N | N | Y | N | Unsigned32 |
| RTP-Called-Max-Jitter-FS1 | 74 | N | N | Y | N | Unsigned32 |
| RTP-Called-Max-Jitter-FS2 | 75 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Packets-Lost-FS1 | 76 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Packets-Lost-FS2 | 77 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Avg-Jitter-FS1 | 78 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Avg-Jitter-FS2 | 79 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Avg-Latency-FS1 | 80 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Avg-Latency-FS2 | 81 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Max-Jitter-FS1 | 82 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Max-Jitter-FS2 | 83 | N | N | Y | N | Unsigned32 |
| RTCP-Called-Max-Latency-FS1 | 84 | N | N | Y | N | Unsigned32 |

| AVP | ACME Diameter Attribute | Start | Interim | Stop | Event | AVP Type |
|----------------------------------|-------------------------|--------------------------|---------|------|-------|------------|
| RTCP-Called-Max-Latency-FS2 | 85 | N | N | Y | N | Unsigned32 |
| Acme-Packets-Specific-MSRP-STATS | 86 | N | N | Y | N | Grouped |
| MSRP-Calling-Packets-Received | 87 | N | N | Y | N | Unsigned32 |
| MSRP-Calling-Octets-Received | 88 | N | N | Y | N | Unsigned32 |
| MSRP-Calling-Packets-Transmitted | 89 | N | N | Y | N | Unsigned32 |
| MSRP-Calling-Octets-Transmitted | 90 | N | N | Y | N | Unsigned32 |
| MSRP-Called-Packets-Received | 91 | N | N | Y | N | Unsigned32 |
| MSRP-Called-Octets-Received | 92 | N | N | Y | N | Unsigned32 |
| MSRP-Called-Packets-Transmitted | 93 | N | N | Y | N | Unsigned32 |
| MSRP-Called-Octets-Transmitted | 94 | N | N | Y | N | Unsigned32 |
| Stir-VS-Verstat | 108 | Y | Y | Y | Y | UTF8String |
| Stir-VS-Reason | 109 | Y | Y | Y | Y | String |
| Stir-TN-Used- For-AS-VSRequest | 110 | Y | Y | Y | Y | String |
| Stir-Div-Signed-Request | 111 | Y (If generate-start=OK) | Y | Y | Y | String |
| Stir-Div-Verified-Request | 112 | Y | Y | Y | Y | String |
| Acme-FlowIDFS1-R | 113 | Y | Y | Y | N | UTF8String |
| Acme-FlowIDFS2-F | 114 | Y | Y | Y | N | UTF8String |
| Acme-FlowIDFS2-R | 115 | Y | Y | Y | N | UTF8String |
| Acme-FlowType-FS2-F | 116 | Y | Y | Y | N | UTF8String |
| Acme-FlowType-FS2-R | 117 | Y | Y | Y | N | UTF8String |
| Acme-SipHdr-TO | 122 | Y | Y | Y | Y | UTF8String |
| Acme-P-VSA-200 to Acme-P-VSA-230 | 123 to 153 | Y | Y | Y | Y | UTF8String |

AVP Definitions

The following table lists and briefly describes the Acme-Packet-Specific-Extension-Rf AVPs.

| AVP | Definition |
|----------------------------------|---|
| Ingress-Realm | realm of origination |
| Egress-Realm | realm of termination |
| Ingress-TGRP | TGRP received |
| Egress-TGRP | TGRP sent |
| Ingress-Trunk-Context | trunk context received |
| Egress-Trunk-Context | trunk context sent |
| Ingress-Local-Signaling-Address | Signaling address of P-CSCF that received the request from the remote element |
| Ingress-Remote-Signaling-Address | Signaling address of the remote element that sent the request to the P-CSCF |
| Egress-Local-Signaling-Address | Signaling address of P-CSCF that sent the request to the remote element |
| Egress-Remote-Signaling-Address | signaling address of the remote element that received the request from the P-CSCF |
| Ingress-Local-Media-Address | media address of P-CSCF on the originating side |
| Ingress-Remote-Media-Address | media address of the remote element on the originating side |
| Egress-Local-Media-Address | media address of P-CSCF on the terminating side |
| Egress-Remote-Media-Address | media address of the remote element on the terminating side |
| NAS-Port | SIP proxy port or the H.323 stack's call signaling RAS port. |
| Acme-Session-Ingress-CallID | Call ID generated by the originating device. |
| Acme-Session-Egress-CallID | Call ID generated by the SBC to represent a two-way transaction. |
| Acme-Session-Protocol-Type | Signaling protocol used for a particular leg of a session (in the case of IWF, there may be two legs). This attribute contains the signaling protocol type; for example, SIP or H323. |
| Acme-FlowID-FS1-F | Unique identifier for every media flow processed by the SBC, flow-set 1 forward direction. |
| Acme-FlowID-FS1-R | Unique identifier for every media flow processed by the SBC, flow-set 1 reverse direction. |
| Acme-FlowType-FS1-F | Codec that describes the flow, flow-set 1 forward direction: PCMU, PCMA, G726, G723, G728, G729, H261, H263, T38. |
| Acme-FlowID-FS2-F | Unique identifier for every media flow processed by the SBC, flow-set 2 forward direction. |
| Acme-FlowID-FS2-R | Unique identifier for every media flow processed by the SBC, flow-set 2 reverse direction. |
| Acme-FlowType-FS2-F | Codec that describes the flow, flow-set 2 forward direction: PCMU, PCMA, G726, G723, G728, G729, H261, H263, T38. |
| Acme-FlowType-FS2-R | Codec that describes the flow, flow-set 2 reverse direction: PCMU, PCMA, G726, G723, G728, G729, H261, H263, T38. |

| AVP | Definition |
|--|--|
| Ingress-Local-Media-Port | Ingress port portion of address of P-CSCF on the originating side |
| Ingress-Remote-Media-Port | Ingress port portion of media address of the remote element on the originating side |
| Egress-Local-Media-Port | Egress port portion of address of P-CSCF on the terminating side |
| Egress-Remote-Media-Port | Egress port portion of media address of the remote element on the terminating side |
| Acme-Session-Charging-Function-Address | The latest cached copy or the configured ccf-address. |
| Acme-Local-Timezone | Local GMT/UTC time zone that is provisioned on the SBC. |
| Acme-Post-Dial-Delay | Amount of time between session initiation and an alerting event. |
| Acme-SIP-Diversion | SIP Diversion header; communicates to the called party from whom and why a call diverted. |
| Acme-Session-Disposition | Status of the call attempt as it progresses from being initiated (using a SIP INVITE or H.323 Setup message) to being either answered or failing to be answered. |
| Disconnect-Initiator | Initiator of a call disconnect. |
| Terminate-Cause | Reason for session ending (refer to Session Termination session). |
| Acme-Session-Disconnect-Cause | Q.850 cause code value. |
| Acme-SIP-Status | SIP status code for RFC 3326 support. |
| Acme-FlowType-FS1-R | Codec that describes the flow, flow-set 1 reverse direction: PCMU, PCMA, G726, G723, G728, G729, H261, H263, T38. |
| RTP-Calling-Octets-FS1 | RTP total calling octets for stream 1 |
| RTP-Calling-Octets-FS2 | RTP total calling octets for stream 2 |
| RTP-Calling-Packets-FS1 | RTP total calling packets for stream 1 |
| RTP-Calling-Packets-FS2 | RTP total calling packets for stream 2 |
| RTP-Calling-Octets-Transmitted-FS1 | RTP calling octets transmitted for stream 1 |
| RTP-Calling-Octets-Transmitted-FS2 | RTP calling octets transmitted for stream 2 |
| RTP-Calling-Packet-Transmitted-FS1 | RTP calling packets transmitted for stream 1 |
| RTP-Calling-Packet-Transmitted-FS2 | RTP calling packets transmitted for stream 2 |
| RTP-Calling-Packets-Lost-FS1 | RTP calling packets lost for stream 1 |
| RTP-Calling-Packets-Lost-FS2 | RTP calling packets lost for stream 2 |
| RTP-Calling-Avg-Jitter-FS1 | RTP calling average jitter rate for stream 1 |
| RTP-Calling-Avg-Jitter-FS2 | RTP calling average jitter rate for stream 2 |
| RTP-Calling-Max-Jitter-FS1 | RTP calling maximum jitter rate for stream 1 |
| RTP-Calling-Max-Jitter-FS2 | RTP calling maximum jitter rate for stream 2 |
| RTCP-Calling-Packets-Lost-FS1 | RTCP calling packet lost rate for stream 1 |
| RTCP-Calling-Packets-Lost-FS2 | RTCP calling packet lost rate for stream 2 |
| RTCP-Calling-Avg-Jitter-FS1 | RTCP calling average jitter rate for stream 1 |
| RTCP-Calling-Avg-Jitter-FS2 | RTCP calling average jitter rate for stream 2 |
| RTCP-Calling-Avg-Latency-FS1 | RTCP calling average latency for stream 1 |
| RTCP-Calling-Avg-Latency-FS2 | RTCP calling average latency for stream 2 |
| RTCP-Calling-Max-Jitter-FS1 | RTCP calling maximum jitter rate for stream 1 |

| AVP | Definition |
|---|--|
| RTCP-Calling-Max-Jitter-FS2 | RTCP calling maximum jitter rate for stream 2 |
| RTCP-Calling-Max-Latency-FS1 | RTCP calling maximum latency rate for stream 1 |
| RTCP-Calling-Max-Latency-FS2 | RTCP calling maximum latency rate for stream 2 |
| RTP-Called-Octets-FS1 | RTP called total octets for stream 1 |
| RTP-Called-Octets-FS2 | RTP called total octets for stream 2 |
| RTP-Called-Packets-FS1 | RTP called total packets for stream 1 |
| RTP-Called-Packets-FS2 | RTP called total packets for stream 2 |
| RTP-Called-Octets-Transmitted-FS1 | RTP called octets transmitted for stream 1 |
| RTP-Called-Octets-Transmitted-FS2 | RTP called octets transmitted for stream 2 |
| RTP-Called-Packets-Transmitted-FS1 | RTP called packets transmitted for stream 1 |
| RTP-Called-Packets-Transmitted-FS2 | RTP called packets transmitted for stream 2 |
| RTP-Called-Packets-Lost-FS1 | RTP called packets lost for stream 1 |
| RTP-Called-Packets-Lost-FS2 | RTP called packets lost for stream 2 |
| RTP-Called-Avg-Jitter-FS1 | RTP called average jitter for stream 1 |
| RTP-Called-Avg-Jitter-FS2 | RTP called average jitter for stream 2 |
| RTP-Called-Max-Jitter-FS1 | RTP called maximum jitter for stream 1 |
| RTP-Called-Max-Jitter-FS2 | RTP called maximum jitter for stream 2 |
| RTCP-Called-Packets-Lost-FS1 | RTCP called packets lost for stream 1 |
| RTCP-Called-Packets-Lost-FS2 | RTCP called packets lost for stream 2 |
| RTCP-Called-Avg-Jitter-FS1 | RTCP called average jitter for stream 1 |
| RTCP-Called-Avg-Jitter-FS2 | RTCP called average jitter for stream 2 |
| RTCP-Called-Avg-Latency-FS1 | RTCP called average latency for stream 1 |
| RTCP-Called-Avg-Latency-FS2 | RTCP called average latency for stream 2 |
| RTCP-Called-Max-Jitter-FS1 | RTCP called maximum jitter for stream 1 |
| RTCP-Called-Max-Jitter-FS2 | RTCP called maximum jitter for stream 2 |
| RTCP-Called-Max-Latency-FS1 | RTCP called maximum latency for stream 1 |
| RTCP-Called-Max-Latency-FS2 | RTCP called maximum latency for stream 2 |
| Acme-MSRP-Calling-Packets-Received | Total MSRP calling packets |
| Acme-MSRP-Calling-Octets-Received | Total MSRP calling octets |
| Acme-MSRP-Calling-Packets-Transmitted | Total MSRP calling transmitted packets |
| Acme-MSRP-Calling-Octets-Transmitted | Total MSRP calling transmitted octets |
| Acme-MSRP-Called-Packets-Received | Total MSRP called packets |
| Acme-MSRP-Called-Octets-Received | Total MSRP called octets |
| Acme-MSRP-Called-Packets-Transmitted | Total MSRP called transmitted packets |
| Acme-MSRP-Called-Octets-Transmitted | Total MSRP called transmitted octets |
| Acme-SipHdr-TO | The TO header of a SIP method, as captured from the sipmsg.log. |
| Acme-P-VSA- <i>NNN</i> (where <i>NNN</i> is 200 to 230) | Empty AVPs reserved for population with header manipulation rules for SIP calls. |

System Alarming Based on Received Result-Code (268) AVP

All non-success (non 2xxx) result codes received are logged and the SBC raises an internal minor alarm, sending the apDiameterSrvrErrorResultTrap SNMP trap to any configured trap receiver for the following values in the ACA message's Result-Code (268) AVP:

- 3002

- 3004
- 4002
- 5012

For more information, see the *MIB Reference Guide*. The SBC uses DWR mechanism for failover purposes - not based on result codes.

The SBC expects this AVP in an ACA message to follow:

| AVP | Number | AVP Type | Reference | Start | Interim | Stop | Event |
|-----------------|--------|------------|-----------|-------|---------|------|-------|
| [Result-Code] | 268 | Unsigned32 | Base | Y | Y | Y | Y |

Interim ACR Message Creation Interval per Acct-Interim-Interval AVP

The ACA message's Acct-Interim-Interval AVP (85) indicates the interval at which the SBC sends INTERIM ACR messages. The value provided in the CCF's ACA overrides any configured Acct-Interim-Interval in the network element.

The SBC expects this AVP in an ACA message to follow:

| AVP | Number | AVP Type | Reference | Start | Interim | Stop | Event |
|-------------------------------|--------|------------|-----------|-------|---------|------|-------|
| [Acct-Interim-Interval AVP] | 85 | Unsigned32 | Base | Y | Y | Y | Y |